Deep Learning

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

Lecture - 56
Image Denoising

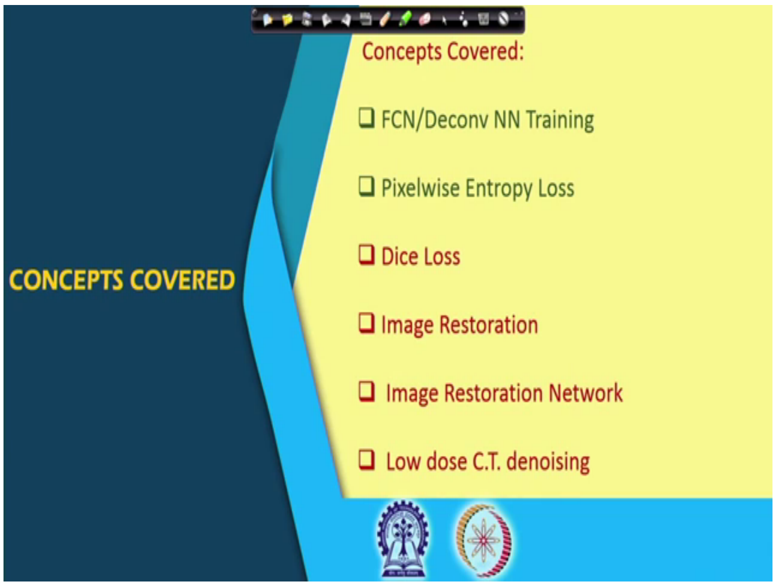
Hello, welcome back to the NPTEL online certification course on Deep Learning. In our previous class we were talking about how to train a fully convolutional neural network or a deep convolutional neural network to be used for semantic segmentation purpose. And we have seen there that; obviously, whenever we want to train a neural network or a deep neural network we need a lot of ground truth data.

So, for training a neural network to be used for semantic segmentation purpose, I need the original input image and the corresponding ground truth which is the semantically segmented image. And when you fed that input image, what I expect is the output of the network will be a segmented image or semantically segmented image. And for training purpose, what we have done is this output map that you get the output map has been represented as an array of one odd vectors.

Where the dimensionality or the dimension of the one odd vector is same as the number of categories of objects which are present within the image. So, if we expect that the image will contain five different objects or five different regions then my one odd vector will be of size five. And you prepare your ground truth in the form of such arrays of one odd vectors. And for training the neural network; obviously, we need to define a loss function.

So, that using the back propagation learning while performing the gradient descent training operation using back propagation learning. Then the loss function the m will be to have a gradient descent algorithm which will try to minimize the loss with respect to the network parameters.

(Refer Slide Time: 02:30)



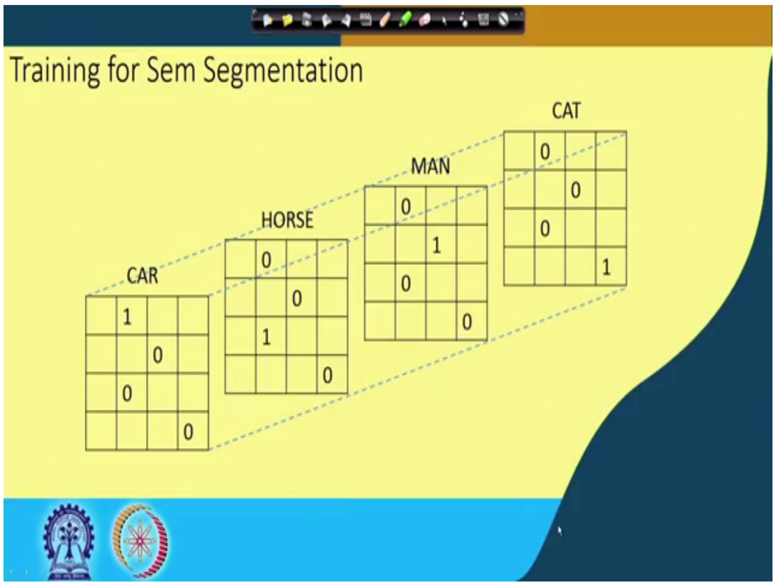
CONCEPTS COVERED

Concepts Covered:

- ❑ FCN/Deconv NN Training
- ❑ Pixelwise Entropy Loss
- ❑ Dice Loss
- ❑ Image Restoration
- ❑ Image Restoration Network
- ❑ Low dose C.T. denoising

So, in your previous class we have talked about a loss function which is pixel wise entropy loss. Today we will talk about another loss which is known as dice loss and after introducing dice loss again, dice loss is the applicable for training the neural network, which is meant for performing semantic segmentation. And then I will move on to another application of the deep neural network, the deconvolution network that is the restoration purpose of our degraded image. And by restoration what I mean is removal or filtering of the noise present in an input image.

(Refer Slide Time: 03:18)

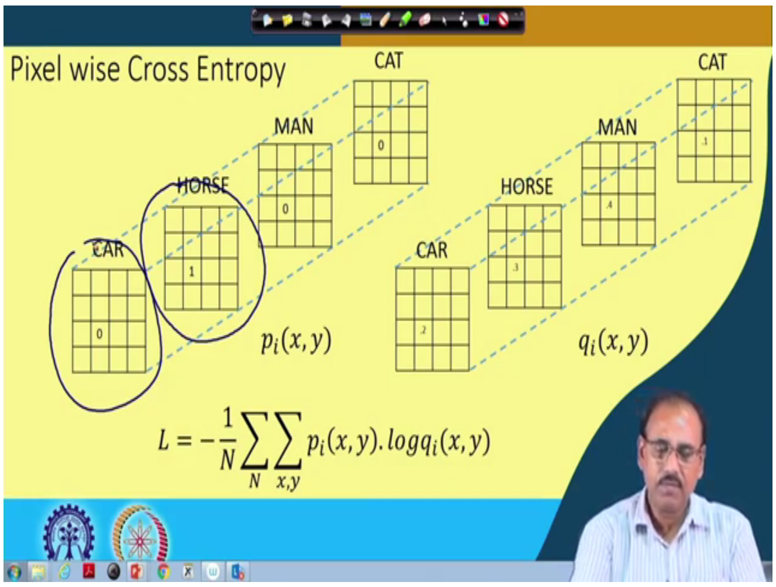


Training for Sem Segmentation

So, let us see that what is dice loss and before coming to that, let me just recapitulate that what is an array of one odd vectors and how using this array of one odd vectors, we have computed the pixel wise entropy loss. So, the array of one odd vectors is something like this, that your output array is of the same size as that of the input image.

But every element in the output array, is now a one odd vector, where if a pixel within the original image belongs to a region occupied by as a horse. Then in the output one odd vector at the corresponding element, the component corresponding to horse will be 1 and all other components will be equal to 0. So, these are the one odd vectors or what are the target vectors.

(Refer Slide Time: 04:14)



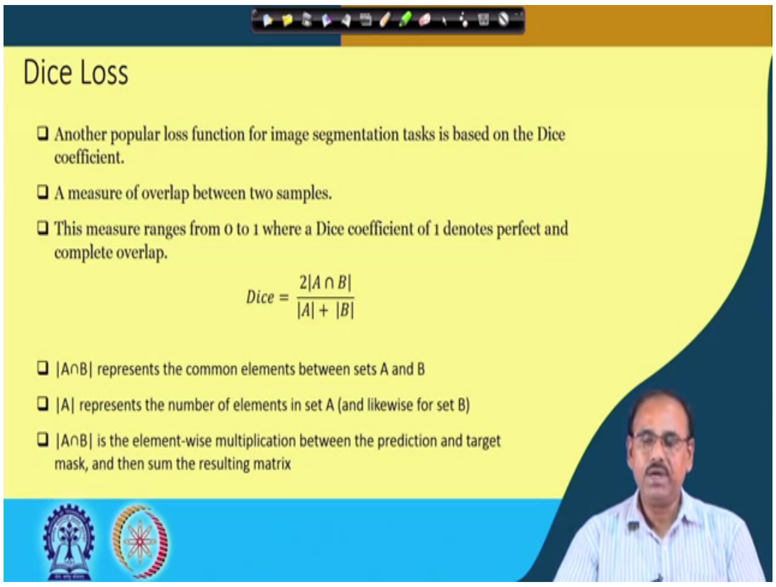
Pixel wise Cross Entropy

$$L = -\frac{1}{N}\sum_{N}\sum_{x,y} p_i(x,y).\log q_i(x,y)$$

But what happens is during training operation your network may actually output something else which is something different from this one odd vector. So, by using your target one odd vector and the actual output vector that you are getting, which is in terms of the probability that the pixel belongs to a particular category. In case of one odd vector it gives you probability equal to 1. That, the component corresponding to horse equal to 1; that means, the probability that the horse of that particular pixel belongs to horse that is equal to 1.

And the probability that it belongs to any other category, that is equal to 0. That is what your one horse vector tells whereas, actually the network gives you some other probability measure. So, there is a difference between your target probability and the actual probability that you get. And using these two you can define a cross entropy, which is basically a measure of difference between your target and actually what you get. And this cross entropy loss can be minimized for training of the neural network.

(Refer Slide Time: 05:25)



Dice Loss

- Another popular loss function for image segmentation tasks is based on the Dice coefficient.
- A measure of overlap between two samples.
- This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap.

$$Dice = \frac{2|A \cap B|}{|A| + |B|}$$

- $|A \cap B|$ represents the common elements between sets A and B
- $|A|$ represents the number of elements in set A (and likewise for set B)
- $|A \cap B|$ is the element-wise multiplication between the prediction and target mask, and then sum the resulting matrix

And the other loss that I was talking about, what is dice loss. The dice loss actually tries to find out that how much is the overlap between the target and actual output that you get. And this overlap is computed over every category or every plane. If I consider this, array of one vectors, array of one odd vectors to be different planes.

So, for example, this is a plane corresponding to car. So, this plane corresponds to car, this plane corresponds to horse and so on. So, if I consider those to be different planes, then in case of this when I try to compute this dice loss what I do is in every plane, I try to find out that what is the amount of overlap between the target and the actual array that you are getting and based on this overlap you try to compute a loss.

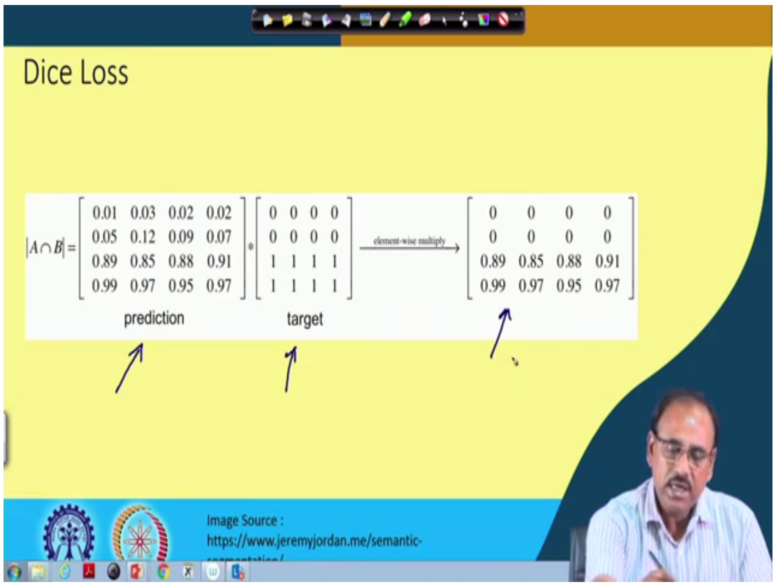
So, that is what is given by what is known as dice coefficient. So, it tries to find out, the overlap which is in the form of A intersection B, where A maybe your target array and B is the actual array that you get. So, if you take A intersection B and mod of that, that gives you the measure of what is the amount of overlap between these two.

And if A and B overlaps exactly, then I will have a maximum value or the maximum cardinality of A intersection B. And if I compute this dice coefficient in that case, obviously; if they overlap exactly then cardinality of A and cardinality B of B will be equal to same and that will also be same as the cardinality of A intersection B.

So, the dice coefficient we have compute in that case, this dice coefficient will be equal to 1 or as if you find, if you compute when there is no overlap, that means A intersection B becomes a null set and the cardinality of that will be equal to 0. Then whatever be the cardinality of A or the cardinality of B, your dice coefficient will be equal to 0. So, in one extreme, there are two extreme cases. In one extreme there is no overlap when the dice coefficient gives you a value 0 and when there is perfect overlap, then the dice coefficient equal to 1.

So, when your target and the actual what you are getting, the matches perfectly your dice coefficient becomes maximum. And when they do not match at all, that dice coefficient becomes 0. So, using this concept we can try to define a loss function. So, in order to find out what is A intersection B, what you can actually do is, you can multiply the two different arrays. One was your target array and the other one is the actual array, in the same category plane and you multiply these two you get the intersection of A and B.

(Refer Slide Time: 08:25)



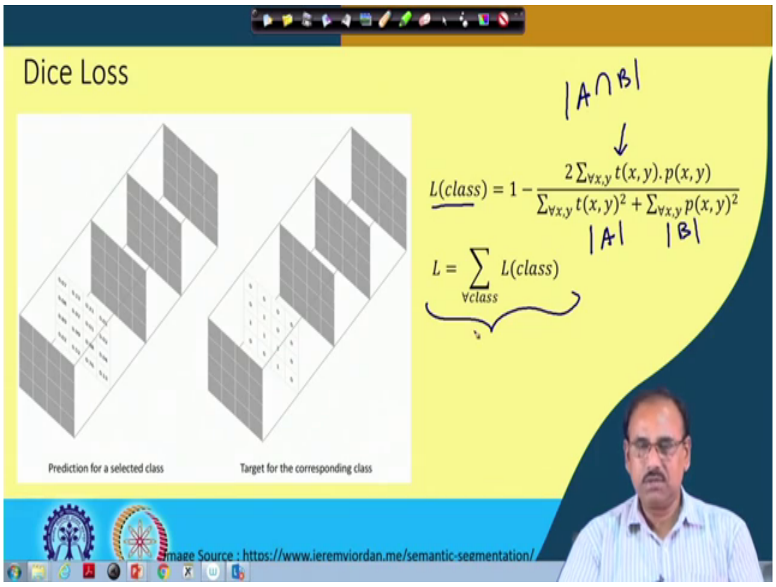
Dice Loss

$$|A \cap B| = \underbrace{\begin{bmatrix} 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.12 & 0.09 & 0.07 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix}}_{\text{prediction}} * \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}_{\text{target}} \xrightarrow{\text{element-wise multiply}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix}$$

Image Source :
https://www.jeremyjordan.me/semantic-

So, that is what has been shown here. So, here A intersection B, you will find that my actual target, if I consider that actual target is this. This is my target array and this is the array that is given by the network. You multiply these two then, your A intersection B becomes this and in order find out the cardinality you take simply take the sum of all the elements in this particular array. At this array has been obtained by point wise or element wise multiplication of the components of A with the corresponding components or elements of B.

(Refer Slide Time: 09:10)



Dice Loss

$$|A \cap B|$$
$$\downarrow$$
$$\underline{L(class)} = 1 - \frac{2\sum_{\forall x,y} t(x,y).p(x,y)}{\underset{|A|}{\sum_{\forall x,y} t(x,y)^2} + \underset{|B|}{\sum_{\forall x,y} p(x,y)^2}}$$

$$\underbrace{L = \sum_{\forall class} L(class)}$$

Prediction for a selected class
Target for the corresponding class

Image Source : https://www.jeremyjordan.me/semantic-segmentation/

And once I define this, then from here I can go for definition of the dice loss. So, dice loss again, because I am computing this for every class. So, dice loss can be computed for every class and then sum over all the classes gives you the overall dice loss. So, here it is defined in this form, as we said that the dice coefficient will be equal to 1 when there is perfect overlap or perfect match and it will be equal to 0 when there is no match. Coming down to the loss function if there is perfect match my loss should be equal to 0 and when there is no match the loss should be maximum.

So, that is how this dice coefficient is computed over here. So, you find that the denominator of this fraction. This corresponds to mod of A intersection B and the denominator corresponds to mod A and mod of B and this is what is actually a dice coefficient. And 1 minus dice coefficient per class gives you the loss within the class. And if I take the sum of loss over all the classes, this gives me the overall segmentation loss and during training using back propagation and gradient descent operation you can try to minimize this loss.

So, your training operation will be such that you modify the network parameters, following the gradient descent, where the gradient of this loss function with respect to parameters will be taken. And you modify the parameters in such a way that this loss function is minimized. So, for training of the neural network which is meant for semantic segmentation either you can use that pixel wise cross entropy loss or I can also use this dice loss for training of the neural network, ok.

Now, given this now I want to move to another application of this deconvolution and neural networks. The application is image restoration or noise filtering. And for doing that what I need to do, what I need to understand is that, what is an image degradation model and why do I need filtering operation. So, let us look at this.

(Refer Slide Time: 11:39)

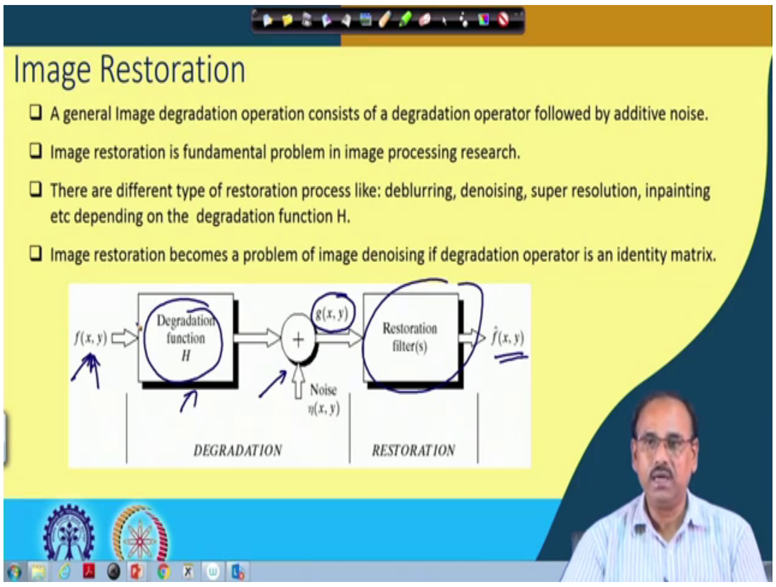


Image Restoration

- A general Image degradation operation consists of a degradation operator followed by additive noise.
- Image restoration is fundamental problem in image processing research.
- There are different type of restoration process like: deblurring, denoising, super resolution, inpainting etc depending on the degradation function H.
- Image restoration becomes a problem of image denoising if degradation operator is an identity matrix.

Any image capturing operation, follows through I mean goes to this degradation steps. So, suppose my actual input image is f x, y. This f x, y goes to a degradation function, which is H as shown over here. So, I have my actual original input image, which is H there is a degradation operator and after degradation operator I have an additive noise. So, at the output of this block, I get an image g x, y, which is degraded and noisiness.

So, the purpose of this distortion is to remove this noise and the degradation. And in classical image processing operations, there are various techniques for performing this task or for restoration; this is what is known as restoration. So, the purpose of restoration is that, I want to have an output image, which is very close to my original input image f x, y. And you find that in this case, if your degradation operator H. This becomes an identity operator, then your degradation model is simply additive noise added to your input image.

So, the image that we get is a noise image and in that case, this restoration operations simply becomes a noise filtering operation, I do not have to consider what is my degradation operator or degradation filter. So, in today's lecture, would; what we will talk about is how we can employ a neural network, a deep neural network or convolutional neural network for performing this image filtering operation or noise filtering operation.

Now, find that when I have discussed about the training of the neural network. You will find that in most of the cases, the architecture of the neural network is same. Whether you use it for classification purpose or whether you use the neural network for segmentation purpose or you use the neural network for even noise filtering purpose. What makes the functions different is how you train your neural network. That means, how you generate your ground truth images or this images that is meant for supervised training of the deep neural network and how you define your loss function.

Your training algorithm is same, it is the same back propagation algorithm using gradient descent. So, what are your training sets and what are and what is your loss function. You these two actually decides that what will be the function of the neural network, how the neural network will behave. So, let us see that how you can use the neural network for noise filtering operation.

(Refer Slide Time: 14:48)

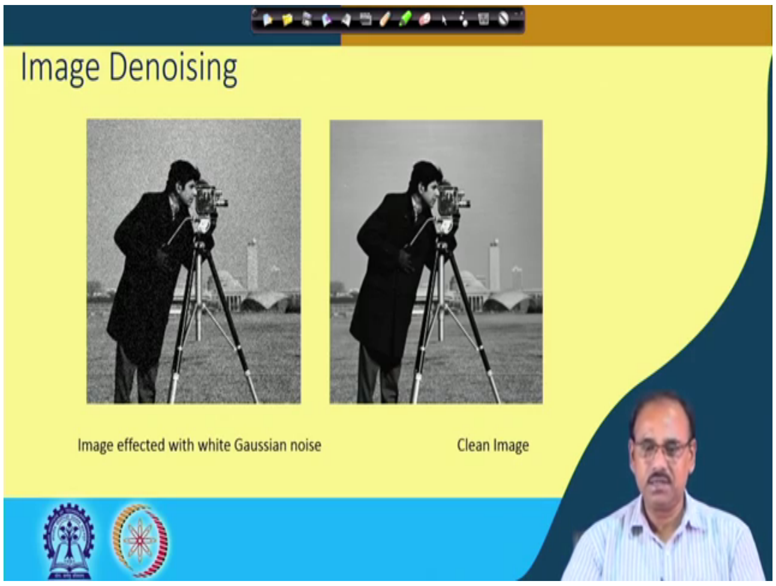


Image Denoising

Image effected with white Gaussian noise

Clean Image

So, by noise filtering what I mean is, here on the input on the left hand side we have an image which is corrupted with additive Gaussian noise. So, you find that what is the nature of the image or what is the characteristic of the image that is on the left hand side which is a noisy image.

By filtering, what I mean is I want to remove this noise, so that I get a clean image and the clean image as shown on the right hand side; obviously, the image that you have on the right hand side, that is much better than your noisy image what you have on the left. So, I want that my neural network should perform this similar task that is this noise filtering operation.

(Refer Slide Time: 15:35)

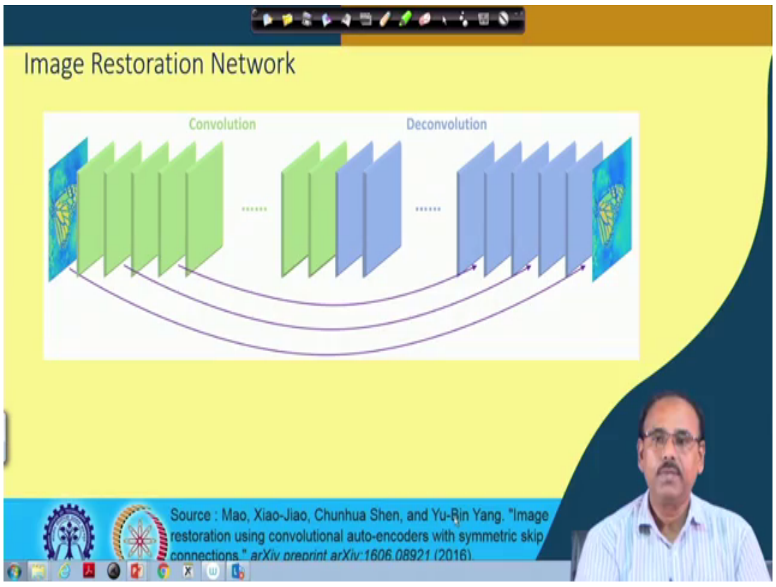


Image Restoration Network

Source : Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." arXiv preprint arXiv:1606.08921 (2016).

So, let us see how we can do it. So, a neural network which has been suggested as suggested by the authors of this paper, which is on image distortion network. So, this image distortion network, it is a deconvolution network, where I have arrays of the convolution networks followed by an arrays of a set of deconvolution networks. And this particular network, the deconvolution network also makes use that the concept of the skip connection.

That is from a convolution layer, you can directly feed the output of the convolution layer to the corresponding deconvolution layer, the pair deconvolution layer in the deconvolution part. And these are the skip connections that is shown over here. So, these are the skip connections that you can have. So, you find that from the convolution layer, you can have a skip connection to the deconvolution layer. So, this is the in general the architecture of the network that is used for image distortion or noise filtering purpose.

(Refer Slide Time: 16:49)

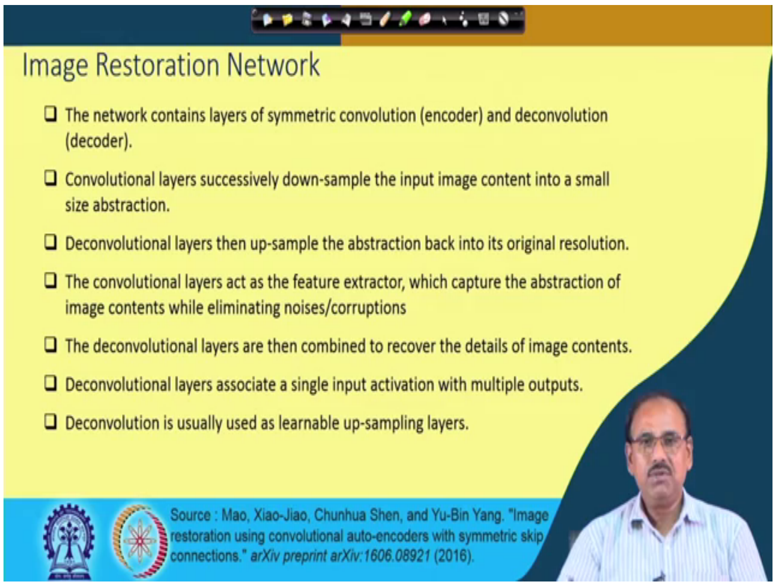


Image Restoration Network

- The network contains layers of symmetric convolution (encoder) and deconvolution (decoder).
- Convolutional layers successively down-sample the input image content into a small size abstraction.
- Deconvolutional layers then up-sample the abstraction back into its original resolution.
- The convolutional layers act as the feature extractor, which capture the abstraction of image contents while eliminating noises/corruptions
- The deconvolutional layers are then combined to recover the details of image contents.
- Deconvolutional layers associate a single input activation with multiple outputs.
- Deconvolution is usually used as learnable up-sampling layers.

Source : Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." *arXiv preprint arXiv:1606.08921* (2016).

Now, what does this convolution layers and the deconvolution layers that actually do. You find that in this particular case, the convolution and deconvolution they have mirror symmetry right. The number of convolution layers and the number of deconvolution layers they are same. The way it is reduced on the convolution side, just in the inverse way the size is increased in the deconvolution side.

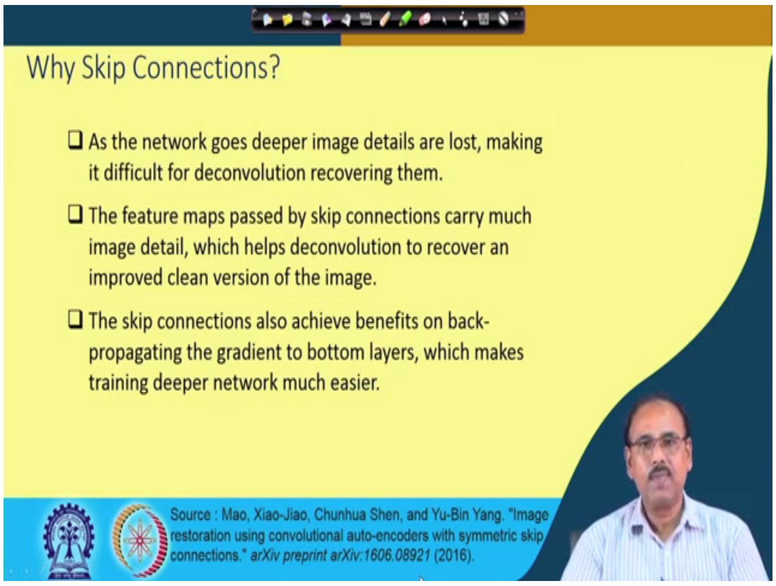
Now, what does this convolution layer do? As we have seen before that, when you perform a convolution operation the convolution operators tries to extract some abstract information or the features of the input image. And when you perform the deconvolution, the deconvolution tries to up sample these abstract information which is collected by the convolution layers and by up sampling this information is being restored back to its original resolution.

So, the operation of the convolution layer and the operation of the deconvolution layer, they are just opposite. So, the convolution layer acts as the feature extracted. And while it tries to extract the features in the process, it eliminates the noise or the corruptions, which are present within the input image. And when you perform the deconvolution operation as the noise has been removed or has been suppressed by the convolution operator.

When I try to deconvolve or up sample the input image, I restored the original size try to restore the original resolution of the image. But, because the noise was suppressed by my convolution operator, when I restored the original resolution, it will be then the noise will be removed from that original revolution. So, you find that when you perform the convolution operation along with the removal of the noise. The convolution operator will also remove certain features or the detailed information, which is present in the image.

So, as a result; when you perform the deconvolution operation, your original restored image or the original resolution image though it will be free of the noise. At the same time, because the convolution layer has removed finer details of the image those finer details will be also be lost from your restored image. And that is where your skip connection plays a very very important role.

(Refer Slide Time: 19:51)



Why Skip Connections?

- ❑ As the network goes deeper image details are lost, making it difficult for deconvolution recovering them.
- ❑ The feature maps passed by skip connections carry much image detail, which helps deconvolution to recover an improved clean version of the image.
- ❑ The skip connections also achieve benefits on back-propagating the gradient to bottom layers, which makes training deeper network much easier.

Source : Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." *arXiv preprint arXiv:1606.08921* (2016).

So, let us see over here; what this skip connection does. As we have seen that because the skip connection, output of a convolution layer is directly added to a deconvolution layer, the corresponding deconvolution layer. The convolution layers, which are in the shallower part, for them the removal of the detailed information of the image will not be that severe. But, as you move inside the deeper layers, the detailed information will be lost more and more.

So, using this skip connections what you are doing is, when you are trying to deconvolve the image or trying to restore its original resolution. You are actually skipping all the intermediate layers and you are getting the detailed information of the corresponding convolution layer and feeding it directly to the deconvolution layer.

So, when the deconvolution layer tries to up sample the image, it will get that detailed information from its pair convolution layer. So, in the up sampled image it will try to retain that information, which is which it receives from the pair convolution layer. That is how, the detailed information, in your restored image will also be maintained.

The skip connection also helps in the back propagation algorithm, that we have seen earlier; that when the network becomes very deep or the number of layers is very large. In that case, what happens is that; it suffers from a problem which is known as vanishing gradient problem. So, this skip connection that helps to solve or to avoid that vanishing gradient problem during the training of the neural network.

(Refer Slide Time: 21:51)

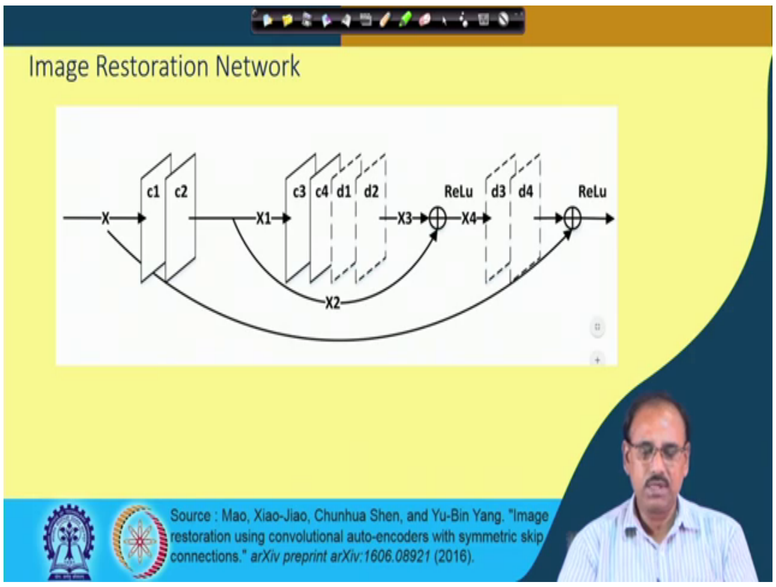


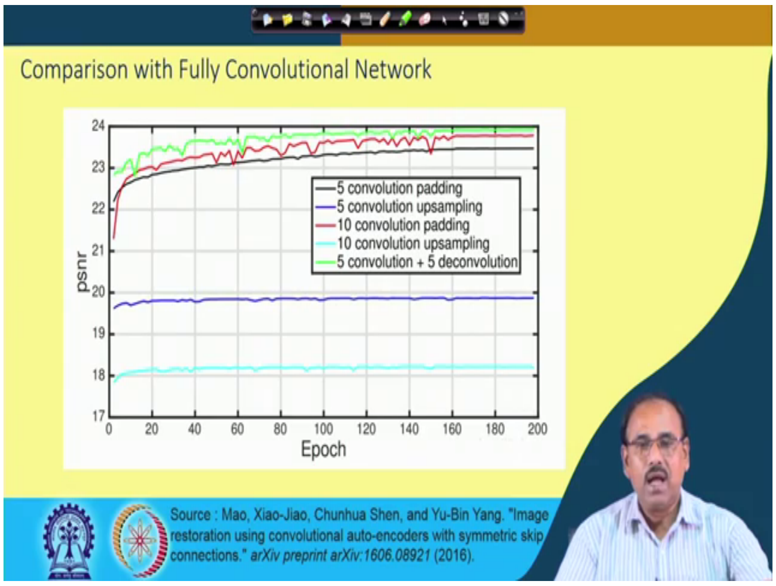
Image Restoration Network

Source : Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." *arXiv preprint arXiv:1606.08921* (2016).

Now, when you have this deconvolution network, the authors of these papers, they have suggested the architecture of the image restoration network or the block of the image restoration network is something like this. So, you find that I have a deconvolution layer say d 4 and the corresponding convolution layer is c 1. Corresponding to c 1 which is a convolution layer, I have the corresponding deconvolution layer d 3.

So, I can have a skip connection from c 1 to d 4, I can have a skip connection from c 2 to d 3 and so on. So, when I have this skip connection, the skip connection is in between the pair layers of the corresponding layers.

(Refer Slide Time: 22:33)



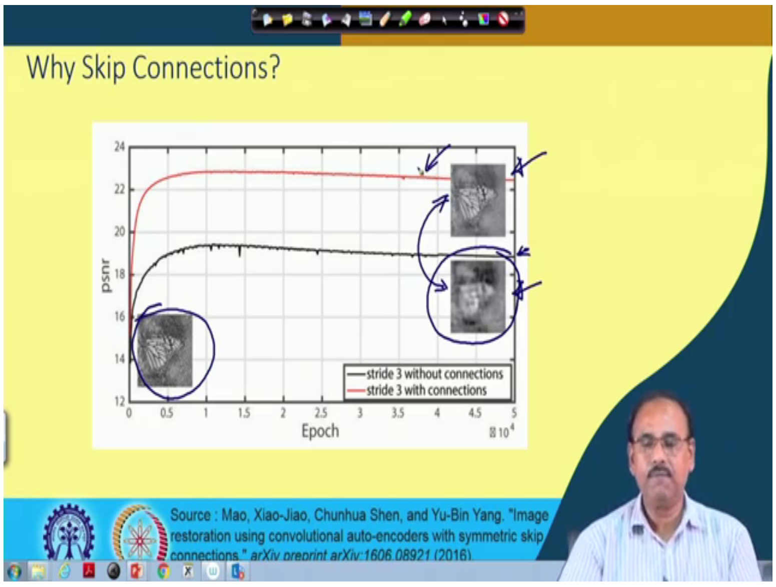
Comparison with Fully Convolutional Network

Source : Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." *arXiv preprint arXiv:1606.08921* (2016).

The authors have also compared, the output compared the performance of their network with some other networks, say the networks which are fully convolutional networks. So, they have compared their result, their performance with fully convolutional networks, having 5 convolution layers and 5 convolutional upsampling layers. A network with 10 convolutional layers and 10 convolution up sampling layers.

And the authors have suggested, a deconvolution layer with deconvolution network with 5 convolution layers and 5 deconvolution layers. And as its shown in this plot, that the performance of this deconvolution network with 5 convolution layer and 5 deconvolution layers is much better compared to competing fully convolution layers. It is better in terms of psnr of the restored image.

(Refer Slide Time: 23:45)



So, the other application and here you find, that how this skip connection network improves the performance or improves the quality of the reconstructed image. Here you find that, over here what we have is the noisy image and this is a restored image, where skip connection was not used. So, here you find that as we said as we said before that, skip connection actually feeds the details information of the image, which are available in shallower layers of the convolution part to its countered deconvolution network deconvolution layer.

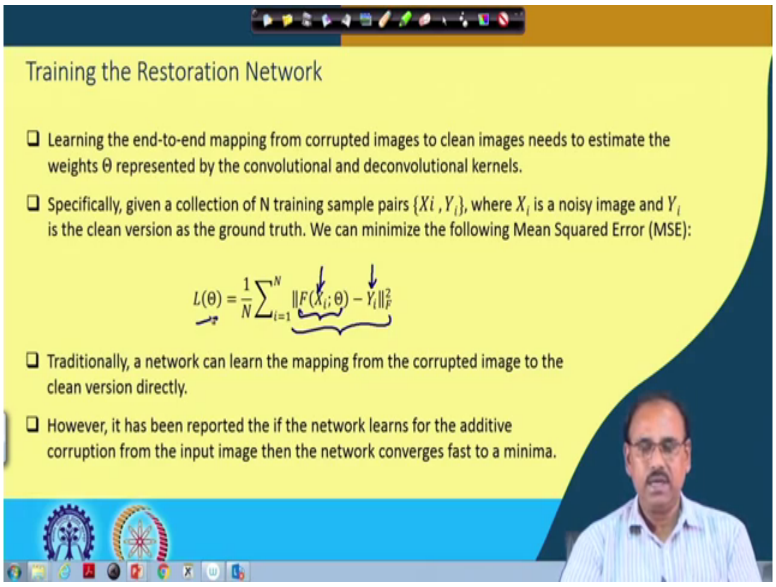
So, that when the deconvolution layer tries to restored the image to its original resolution, the detailed image information we also be available. And if I do not use such skip connections, this detailed image information will be lost and that is what has been shown over here. So, this is the output without skip connection and this is the output that you get with skip connection.

And if you compare these two restored images, it is quite obvious; that the quality of the image that you get with skip connection is much better than the quality of the image that you get without skip connection, simply convolution deconvolution layers but, there is no skip connection operation.

And it is also visible in terms of improved psnr. You will find that without skip connection the psnr is very poor and when you use the skip connection, the psnr value is quite large. So, this skip connection helps both for improvement of the quality of the restored image as well as it helps in training of the neural network. Now again, the most important part as we said, that how do you train this neural network. And as we said that for training of this neural network, I need that ground truth; that means, I have to have an input noisy image and I have to have the corresponding clean image.

So, the way you try to train your neural network is you feed the input to the neural network, at the output of the deconvolution layer at the final output layer, you get the reconstructed image of the restored image. I want that restored image should be same as the clean image and this noisy and clean image pair is used for training of this neural network. So, you compute what is the error between the reconstructed image and the clean image and using back propagation and gradient descent, you would try to minimize this error.

(Refer Slide Time: 26:37)



Training the Restoration Network

- Learning the end-to-end mapping from corrupted images to clean images needs to estimate the weights Θ represented by the convolutional and deconvolutional kernels.
- Specifically, given a collection of N training sample pairs $\{X_i, Y_i\}$, where X_i is a noisy image and Y_i is the clean version as the ground truth. We can minimize the following Mean Squared Error (MSE):

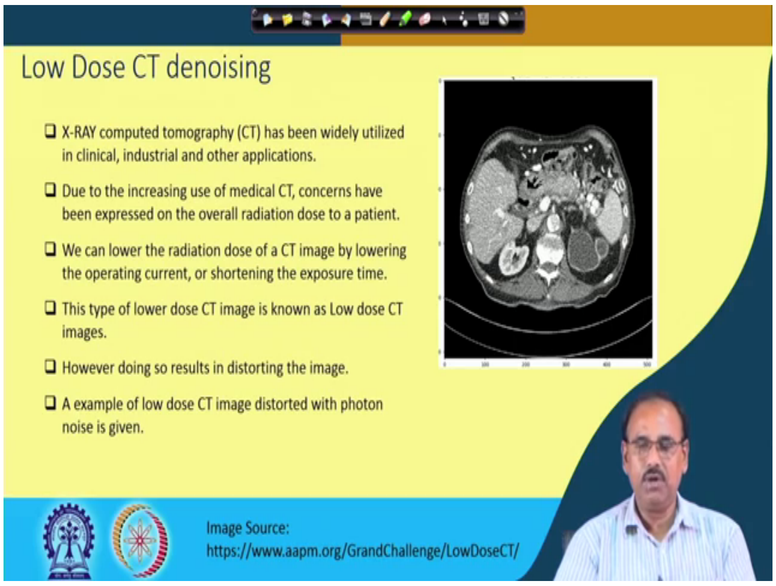
$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\|\underbrace{F(X_i;\Theta)} - Y_i\|_F^2}$$

- Traditionally, a network can learn the mapping from the corrupted image to the clean version directly.
- However, it has been reported the if the network learns for the additive corruption from the input image then the network converges fast to a minima.

So, the error function in this case, will be something like this. So, my input pair is X i Y i, where X i is the noisy image and Y i is the clean image. And what my network gives is the function of X i and theta, where theta is the network parameter. And my loss function is F X i theta minus Y i, Y where Y i is the clean image and this F X i is the restored image, restored by the neural network, X i being the noisy image and Y i is the corresponding clean image. You compute what is the error between these two.

And some of these errors over all the training images that gives you the loss function which is L theta or theta is the network parameter. And then you try to minimize this loss function, iteratively using a gradient descent approach, back propagation approach to update the network parameters theta and that is how the network is to be trained. And once the network is trained then give in an input noisy image, the network will give you an output image which is a clean image of the noise filtered image.

(Refer Slide Time: 28:02)



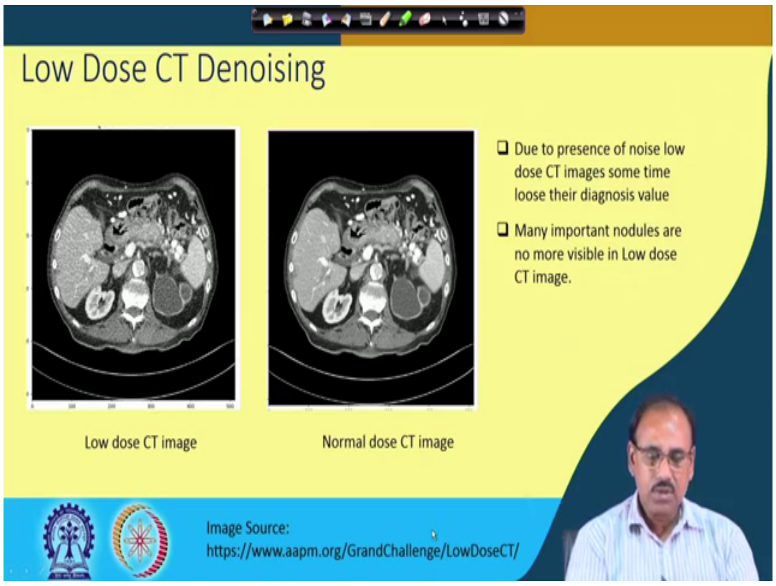
Low Dose CT denoising

- X-RAY computed tomography (CT) has been widely utilized in clinical, industrial and other applications.
- Due to the increasing use of medical CT, concerns have been expressed on the overall radiation dose to a patient.
- We can lower the radiation dose of a CT image by lowering the operating current, or shortening the exposure time.
- This type of lower dose CT image is known as Low dose CT images.
- However doing so results in distorting the image.
- A example of low dose CT image distorted with photon noise is given.

Image Source: https://www.aapm.org/GrandChallenge/LowDoseCT/

There are other applications as has been suggested over the same approach is in low dose CT denoising. You find that when you compute, when you take the X ray CT images, if you increase the X ray energy or the dose then your image will become clean. But if you reduce the radiation energy, the image will be noisy as its shown over here. So, you can use the similar technique, similarly denoisy noising technique using the deep convolution deconvolutional networks for cleaning these images.

(Refer Slide Time: 28:38)



Low Dose CT Denoising

- Due to presence of noise low dose CT images some time loose their diagnosis value
- Many important nodules are no more visible in Low dose CT image.

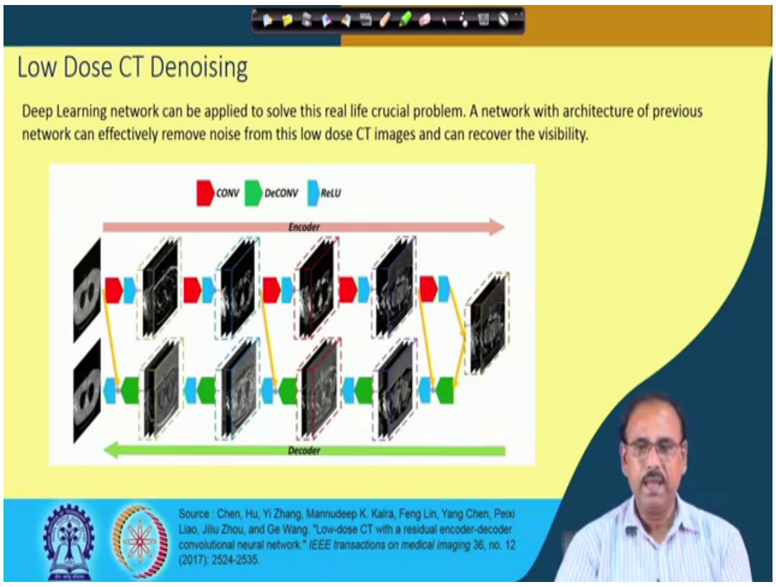
Low dose CT image

Normal dose CT image

Image Source: https://www.aapm.org/GrandChallenge/LowDoseCT/

So, this example shows that again, for training purpose I need the clean low dose image and the high dose pair. Clean image and noise image pair for training of the network.

(Refer Slide Time: 28:52)



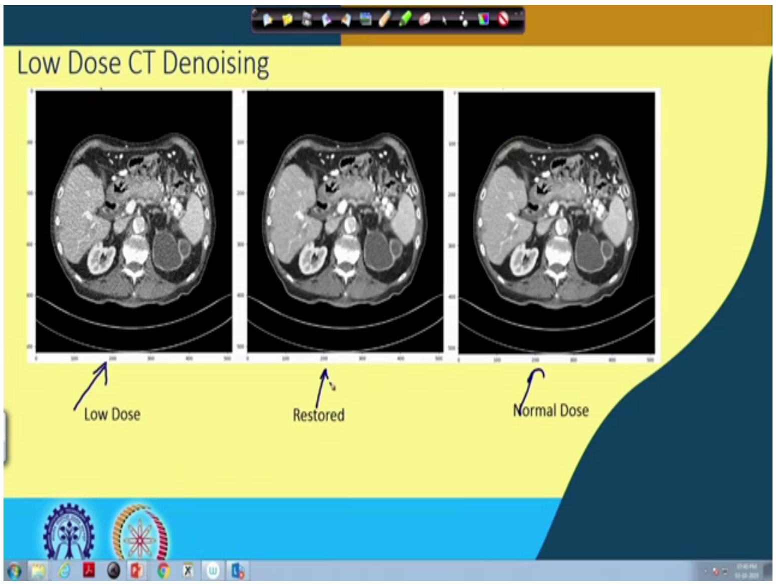
Low Dose CT Denoising

Deep Learning network can be applied to solve this real life crucial problem. A network with architecture of previous network can effectively remove noise from this low dose CT images and can recover the visibility.

Source : Chen, Hu, Yi Zhang, Mannudeep K. Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. "Low-dose CT with a residual encoder-decoder convolutional neural network." *IEEE transactions on medical imaging* 36, no. 12 (2017): 2524-2535.

And once the network is trained and this is the architecture of the network that can be used for this denoisy purpose; denoising purpose, which is same as that skip connection network.

(Refer Slide Time: 29:03)


Low Dose CT Denoising

And this is one of the an example result, that on the left hand side what you have is a low dose ct image. This is the normal CT image and the middle what you have a CT image, which is restored by a trained in deep neural network. The advantage is that, if you have a low dose image then many of the low dose in the image may be missing, which may create a problem for diagnosis based on the CT images.

So, once you have a cleaned image then the diagnosis becomes much more reliable. So, in today's lecture we have talked about one of the loss functions that is that dice loss, which can be used for training a neural network meant for the semantic segmentation purpose. And we have talked about another application of this deep neural network, which is for denoising of the noisy image.

So, with this we will stop our discussion on the discriminative networks. In our next class, we will start talking about the generative network as we said that your deep learning consider is there are two different aspects. One is for classification aspect, which is the discriminating network and the other one is generating aspect, which are done by generative network or generative learning. So, in our next class we will talk about how we can have a generative neural network.

Thank you.