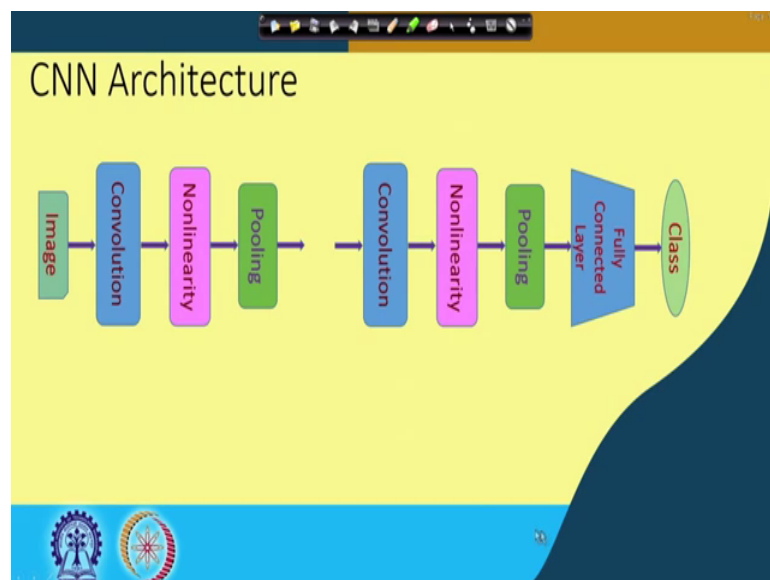


Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture – 37
MLP versus CNN, Popular CNN Architecture: LeNet

Hello. Welcome to the NPTEL, online certification course on Deep Learning. So, we are now talking about the convolutional neural network and today we will talk about some of the popular convolutional neural network models.

(Refer Slide Time: 00:47)



Now, before we go for this let us see a brief recapitulation of what we have done in the previous class. So, in our previous class, we have discussed about the building blocks of a convolutional neural network. So, we have seen that at the heart of the network, there is a layer called convolutional layer, and what this convolutional layer does is it performs the convolution of the input image if it is at the input layer or if it is any in any of the hidden layer then it performs the convolution of the feature map that is available from the previous layer.

And then this convolutional nodes, every node in the convolutional layer that has a non-linear activation function. We have of course, discussed about the power of non-linearity that non-linear mapping can map non-linearly, non-separable data points into linearly separable data points which helps in classification later on. So, we have this convolution

layer followed by the non-linear activation function and then we have another layer which is called pooling layer. In some cases they are also called sub sampling layer. And we have seen that we can have different types of pooling like max pooling or average pooling and so on. And, we said in the previous lecture that the basic purpose of the pooling layer is to collect the local statistic of the feature map and also it gives you the reduction the dimension reduction of the feature maps.

So, this convolution non-linearity followed by pooling we can have different stacks of these three building blocks to give you a deep convolutional neural network. And towards the output side what we have is one or more connected fully connected layers. So, the purpose of using these fully connected layers is to go for classification of the input data that you feed. So, what this convolutional layers?

Convolution non-linearity and pooling these three taken together they compute the feature maps, and then fully connected layer that classifies the input data that is based on the feature map generated by all the previous layers. So, these are the basic building blocks of a convolutional neural network.

(Refer Slide Time: 03:31)

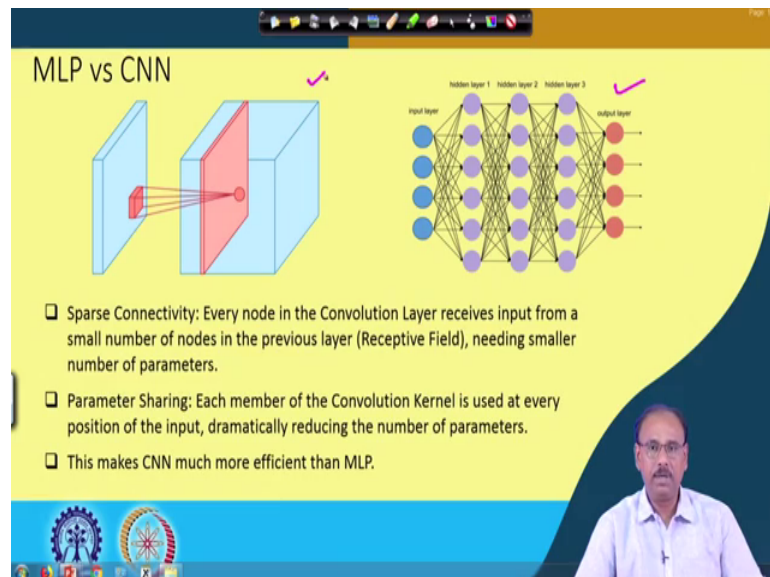


And then we had also seen the structure of a typical CNN architecture. So, here it shows that at the input we have the given image, then the image passes to the convolutional layer and normally the non-linearity which is used in convolutional neural network is known as ReLU or rectified linear unit. Unlike, what we have discussed in the previous

case in case of MLP that many of the cases the non-linearity which is used is sigmoidal function or tan hyperbolic function, but in case of CNN normally the kind of non-linearity which is used is ReLU. Then followed by that you have the pooling layer, where we said the purpose of the pooling layer is to collect the local statistic of the feature map and simultaneously it also reduces the dimension of the feature map.

So, you have a number of such combinations that is convolution, non-linearity and pooling and as has been shown towards the end we have one or more fully connected layers which performs the classification of the input data.

(Refer Slide Time: 04:52)



So, given this now we can compare the architectures of multi-layer perceptron against the convolutional neural network. So, you find over here that on the right hand side, you have been given the structure of a multi-layer perceptron and on the left hand side what you have is the structure of a convolutional neural network. So, in case of multi-layer perceptron, every node in any layer gets input from every other node in the layer before it.

So, every node in say k th layer feeds input to every node in $k+1$ th layer, where so obviously, the connectivity in multi-layer perceptron is quite dense. Against that, if you look at the multi-layer perceptron in case of multi-layer perceptron the connectivity is quite sparse. The reason being every node in the convolution layer receives input from a small number of nodes in the previous layer which we said earlier that these nodes in the

previous layer they are typically called the receptive field. So obviously, as the number of connections to any node in the convolution layer comes from only a limited number of nodes in the previous layer, the number of connections in this case or the connectivity in case of convolutional neural network will be quite sparse or that will require very smaller number of parameters.

The other advantage that you can see in case of CNN network is what is known as parameter sharing. So, the parameter sharing is that each member of the convolution kernel what we said is a convolution kernel typically of size say 3 by 3 or 5 by 5 and so on. Every convolution kernel has got a number of coefficients or parameters or weights whichever you call it and each member of this convolution kernel is used for every pixel or every position in the input feature map.

So, as we said that if this convolution is on the input image then it will be used every member of the convolution kernel will be used for every pixel in the input image and in case of hidden layers or deeper layers, every member of the convolution kernel will be used for every element of the feature map coming from the previous layer.

So, as a result, the number of parameters that is used in case of convolutional neural network is drastically reduced because in case of the fully connected multi-layer perceptron for every pixel or every element in the previous layer I have dedicated parameter, ok. So, for every pixel in the previous layer, if the number of nodes in the next layer is n ; so, every pixel in the previous layer will be feeding input to every node in the next layer.

So, recurring n number of parameters for every pixel in the previous layer for which in case of convolution neural network is much less because every element of the convolutional kernel is used for every location in the feature map which has been fed into this convolutional layer. So, that makes the CNN much more efficient than the multi-layer perceptron because as the number of parameters is drastically reduced the amount of memory that will be required to save these parameters at the same time the number of computations that will be required is greatly reduced.

So, these are the advantages of the convolutional neural network over multi-layer perceptron. So, given this, let us now try to see or discuss about some of the popular CNN models.

(Refer Slide Time: 09:18)

LeNet 5

- Proposed by Yann LeCun, Leon Bottou, Yosuha Bengio and Patrick Haffner for handwritten and machine-printed character recognition.
- Used by many Banks for recognition of hand written numbers on cheques.
- This architecture achieves an error rate as low as **0.95%** on test data

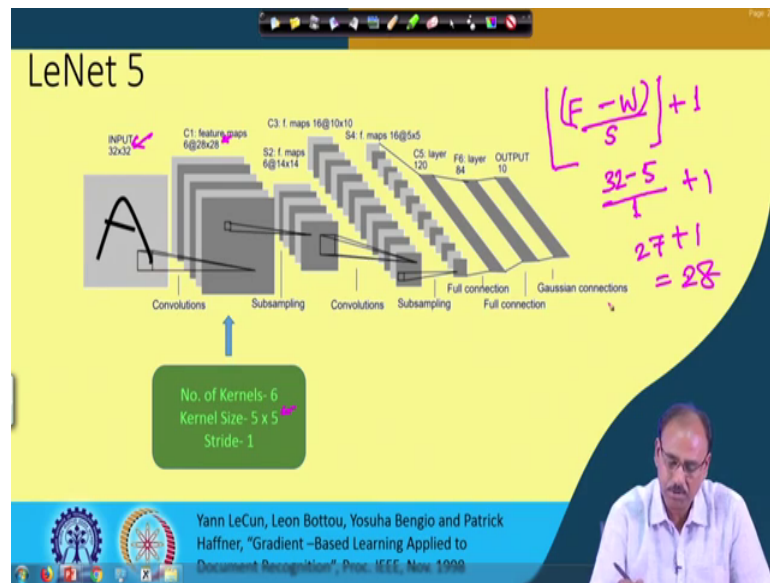
The diagram illustrates the LeNet 5 architecture. It starts with an **INPUT 32x32** image of the digit 'A'. This is processed through two convolutional layers: **C1 feature maps 6@28x28** and **S2 1 maps 6@14x14**. The S2 layer is a subsampling layer. This is followed by another two convolutional layers: **C3 1 maps 16@10x10** and **S4 1 maps 16@5x5**. The S4 layer is another subsampling layer. The output of S4 is connected to a fully connected layer **C5 layer 120**. This is followed by another fully connected layer **F6 layer 84**. The final output is **OUTPUT 10**, which is connected to **Gaussian connections**. The diagram also labels the operations: **Convolutions** for the C1, C3, and C5 layers, and **Subsampling** for the S2 and S4 layers.

Yann LeCun, Leon Bottou, Yosuha Bengio and Patrick Haffner, "Gradient-Based Learning Applied to Document Recognition", Proc. IEEE, Nov. 1998

So, the first model that we will talk about is what is known as LeNet. So, what is this LeNet? LeNet was proposed sometimes in late 90s in 1998 by Yann LeCun, Leon Bottou, Yosuha Bengio and Patrick Haffner, and the purpose of linear LeNet was to recognize handwritten and machine printed characters. And it was used by many banks at that time for the recognition of written numbers on cheques. So, you find that it was widely used for banking automation.

And because it was used for the automation of reading the cheques, naturally the accuracy that was demanded was quite high and you find that LeNet which was having say 5 layers of operations, it could deliver an error rate which was even as low as 0.95 percent on the test data; that means, the accuracy on the test data was more than 99 percent which is quite impressive. So, at the bottom of this slide we show the typical the architecture of this LeNet 5 model.

(Refer Slide Time: 10:40)



So, what does this LeNet 5 model has? You find that the input or to this neural network to this convolutional neural network was grayscale images of size 32 by 32. So, if the input images was of size more than that it had to be scaled down to size 32 by 32. Similarly, if the input image size was less than this actual image size is less than this then it has to be scaled up to size 32 by 32, so that the image is accepted by LeNet.

And then you had a convolution layer which in this case is shown as layer C 1 using 6 kernels every kernel was of size 5 by 5 with and the convolution was performed with stride equal to 1. So, as a result because there are 6 kernels kernels, so this convolution layer generates 6 different feature maps and the kernel size was 5 by 5 with stride one, so that tells you that every feature map will be of size 28 by 28.

So, you remember that your input image was of size 32 by 32 and every feature map which is generated by convolution layer in case of LeNet that is the first convolution layer that those are of size 28 by 28. So, obviously, no padding was used for this convolutional layer. So, you remember that earlier we said that if we want to have the size of the feature map same as the input in that case we have to have extra rows and columns which are known as padding.

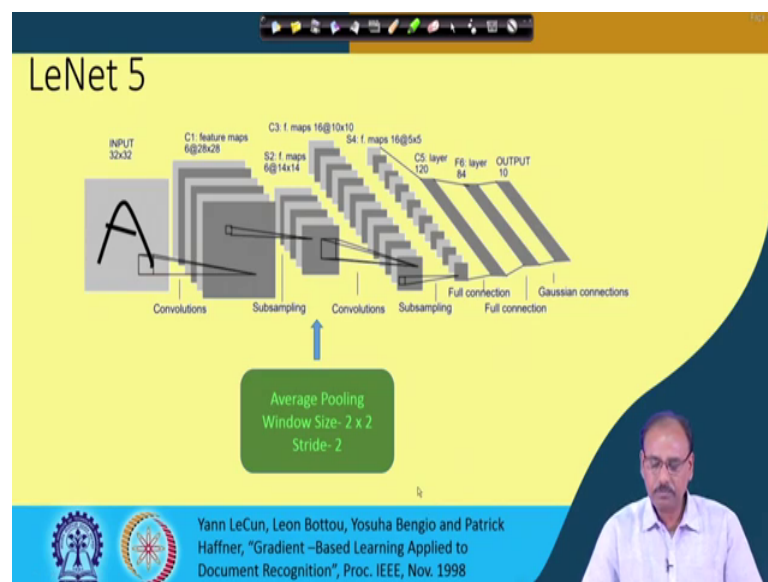
And usually what you do is these extra rows and columns contains element values which is 0. So, that is 0 padding. So, that the size of the output feature map remains same as the size of the input. Now, I will just try to mention in this case that how do you compute

what will be the size of the feature map given the input feature map size and the size of the kernel as well as the stride. So, the formula for doing that if your input feature map is of size say F and you use a convolution kernel of size W , then it is F minus W divided by the stride that you use which is S take the floor function and add 1 to it, that gives you the size of the output feature map.

So, in this case, you find that the input image was of size 32 and the convolution kernel is of size 5. So, it is 32 minus 5 divided by S stride in this case equal to 1, so that gives you this is 27 plus, so this is 27 plus 1 which is equal to 28. So, that gives you the output feature map of size 28 by 28. So, this is how you can find out that what will be the output of the feature map given the input feature map size, the kernel window size and the stride.

The next layer as we said over here. So, this output of this convolution layer passes through a non-linearity which is ReLU and then we have the pooling layer or sub-sampling layer.

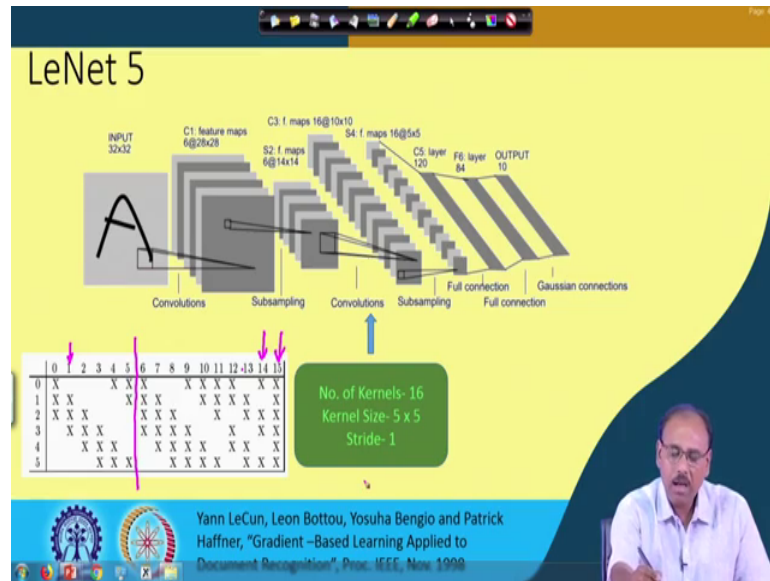
(Refer Slide Time: 14:37)



So, pooling which is used in this case is the average pooling, it is not max pooling it is average pooling with window size 2 by 2 and stride equal to 2. So, after pooling the size of the feature map becomes 14 by 14. Again, following the same formula that we have shown in case of convolution the same formula applies in this case as well. So, you get output feature map size which is 14 by 14. So, that is just half of the input feature map.

So, as we said earlier that pooling reduces the dimension of the feature map and at the same time it collects the local statistic, a local neighborhood statistic or if the number of channels, that remains the same which is equal to 6. So, these inputs, these feature maps of size 14 by 14, these 6 channels or 6 feature maps are again passed to a convolution layer.

(Refer Slide Time: 15:39)



And this time this convolution layer or the second convolution layer in this case it is layer C 3 that has got 16 kernels every kernel is of size 5 by 5 and stride is equal to 1. So, you find that from 6 feature maps you are making 16 different feature maps. So obviously, for that we need 16 kernels, right. And coming to the size of the feature map of each of the 16, again following the same equations you can find the size of the feature maps in this case will be 10 by 10.

Now, when you are using the 16 kernels to generate 16 feature maps, not every 6 feature maps from the input from the previous layer is passed to every kernel, rather it uses a type of asymmetrical connection. So, this particular figure table on the left it tells that how this connection is actually made. So, you find that the 0th kernel for this convolution operation that collects inputs from feature map 0 feature map 1 and feature map 2 of the previous layer.

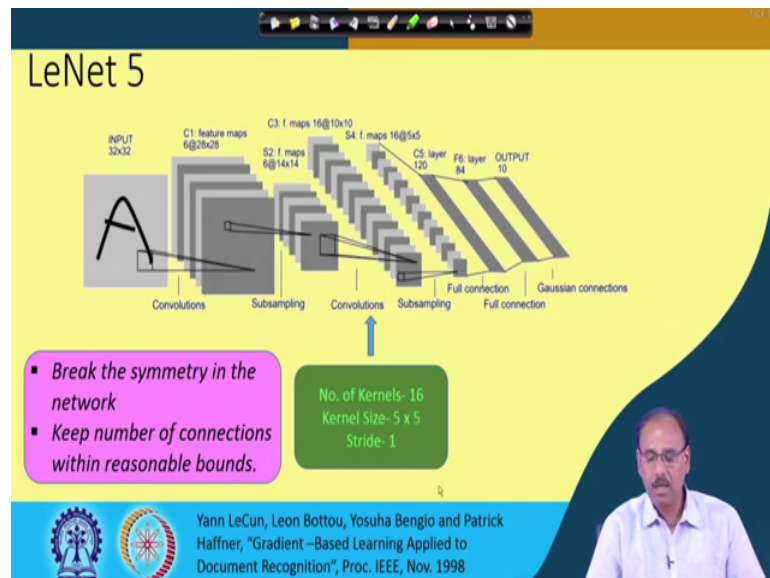
Similarly, the first kernel it gets the input from feature map 1, feature map 2 and feature map 3, second kernel gets inputs from feature map 2, feature map 3 and feature map 4

and so on. So, this way if you continue you find that kernels from 0 to 5 each of these kernels gets input from three consecutive feature maps of the previous layer. In the same manner, sixth kernel receives input from 4 consecutive feature maps of the previous layer.

Seventh kernel gets input from next 4 consecutive inputs from the feature map feature map from the previous layer and that continues up to layer yeah up to layer 11. So, up to this every kernel gets 4, gets input from 4 consecutive feature maps of the previous layer and before that every kernel gets inputs from 3 consecutive inputs on that feature, 3 consecutive feature maps of the previous layer. And then, here you find that layer 14 it gets input from feature map 0, feature map 2, feature map 3 and feature map 4, whereas, kernel 15 gets input from all the feature maps.

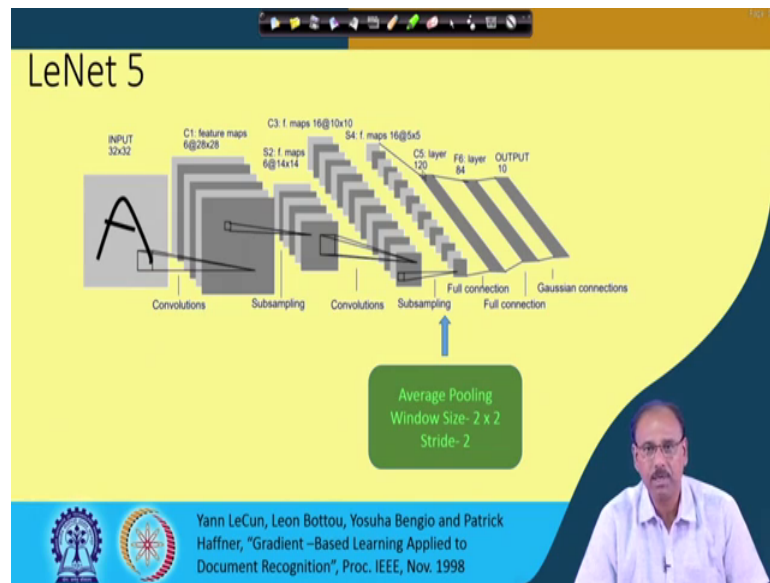
So, the kind of connection or the way the feature map is generated by this convolution layer is a bit asymmetric, it is not symmetric. And the reason, for this making it asymmetric is to break the symmetry in the network that is the first reason and the second is because of this asymmetry, the number of parameters or the number of connections were kept within a reasonable bound.

(Refer Slide Time: 18:57)



So, this was the purpose of having this asymmetric connection to generate this 16 feature maps.

(Refer Slide Time: 19:23)

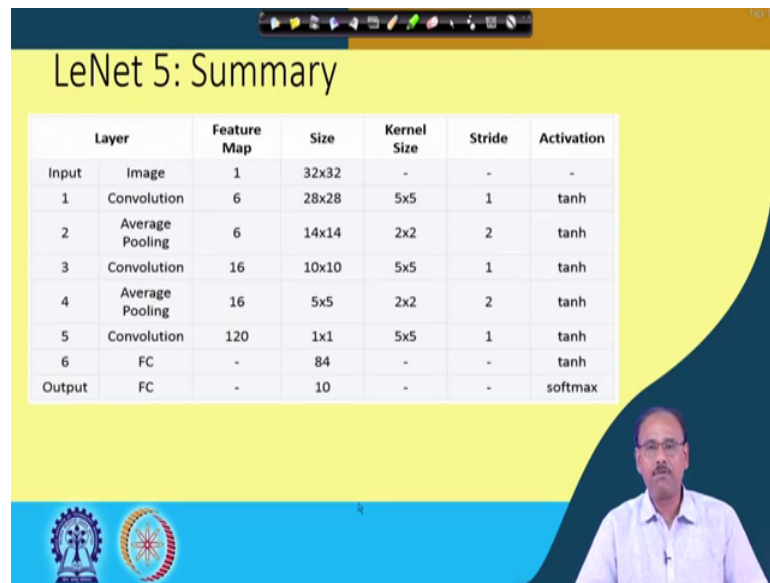


So, after this 16 feature maps again we have a pooling layer or the subsampling layer. So, here again the pooling window was 2 by 2 and with stride equal to 2, that gives you 16 feature maps each of size 5 by 5, ok. And then, what we have is some fully connect fully connected network or fully connected layers and as we said the purpose of such fully connected layers is to classify the input data or to recognize the input data that you have.

So, we have 3 such layers, fully connected layers. For all the layers this fully connected layer C 5 and F 6 the non-linearity which was used was tan hyperbolic whereas, for the output layer the non-linearity which is used is soft max non-linearity or this a soft max classifier. And the number of nodes in the output layer is 10, because this network was used for recognition of numerals in handwritten cheques. So, numerals are from 0 to 9, there are 10 numerals. So, as a result the number of nodes in the output layer is ten which recognizes numerals from 0 to 9.

So, this is what the connection of LeNet 5. And as we said that it was quite successful network for recognizing the input neural numerals the handwritten and machine printed numerals. And it was widely used in the banking sector.

(Refer Slide Time: 21:06)



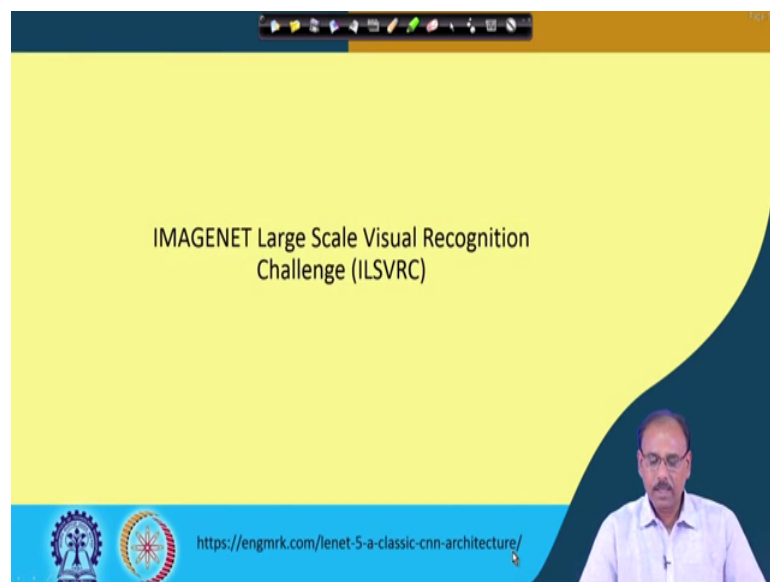
LeNet 5: Summary

Layer	Feature Map	Size	Kernel Size	Stride	Activation	
Input	Image	1	32x32	-	-	
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

The slide features a yellow background with a dark blue curved shape on the right. At the bottom left, there are two circular logos. A video feed of a presenter is visible in the bottom right corner.

So, this next slide it gives you a summary of the connection. So, the summary of the LeNet 5 architecture; so, this was one of the popular neural convolutional neural network model which was used in late 90s for banking automation.

(Refer Slide Time: 21:37)



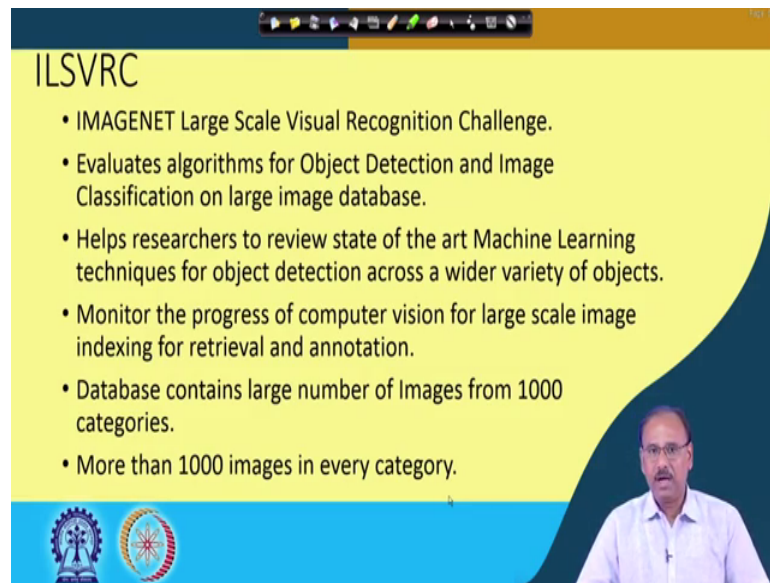
IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)

<https://engmrk.com/lenet-5-a-classic-cnn-architecture/>

The slide has a yellow background with a dark blue curved shape on the right. At the bottom left, there are two circular logos. A video feed of a presenter is visible in the bottom right corner.

Now, before we go for the other neural network models let me tell you about challenge which is known as image net large scale visual recognition challenge or ILSVRC.

(Refer Slide Time: 21:51)



The slide features a yellow background with a dark blue curved shape on the right side. At the top left, the text 'ILSVRC' is displayed. Below it is a bulleted list. In the bottom right corner, there is a small video inset showing a man in a white shirt speaking. At the bottom left, there are two circular logos: one of the Indian Institute of Technology (IIT) Bombay and another of the Indian Institute of Space Science and Technology (IISST).

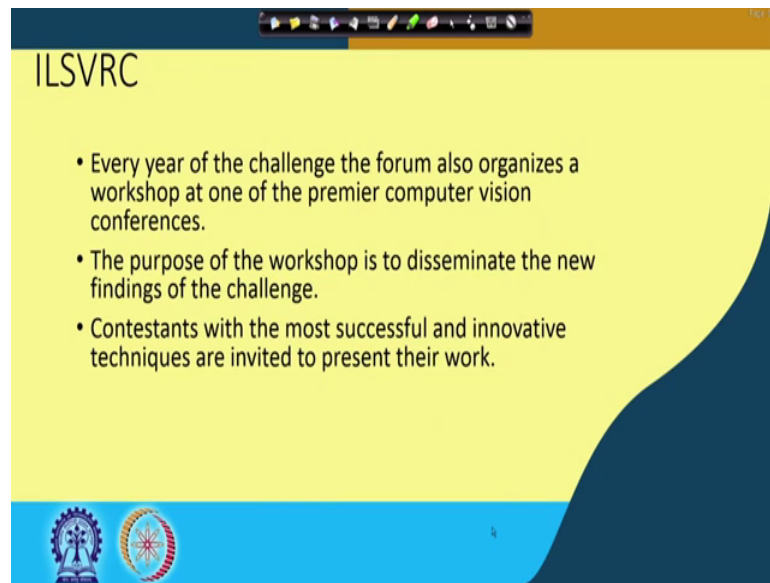
ILSVRC

- IMAGENET Large Scale Visual Recognition Challenge.
- Evaluates algorithms for Object Detection and Image Classification on large image database.
- Helps researchers to review state of the art Machine Learning techniques for object detection across a wider variety of objects.
- Monitor the progress of computer vision for large scale image indexing for retrieval and annotation.
- Database contains large number of Images from 1000 categories.
- More than 1000 images in every category.

So, what is this IMAGENET Large Scale Visual Recognition Challenge? This ILSVRC is a forum that actually evaluates algorithms of object, detection and image classification on large image database. And how does it help? It actually helps the researchers to review and to know state of the art machine learning techniques for object detection across a wide variety of objects. And the researchers can also monitor the progress of computer vision for large scale image indexing for retrieval and automation.

And as the name suggests that it is large scale visual recognition challenge, so the database which is used in this case contains large number of images from 1,000 categories. So, you find that the database is quite huge. And, the recognition or the classification of images on such a database which contains data from 1,000 different categories and more than 1,000 images in every category; so, the database is quite huge. And this is what is given as a challenge known as IMAGENET Large Scale Visual Recognition Challenge.

(Refer Slide Time: 23:24)



So, this challenge is conducted every year. And in every year of challenge, in addition to this challenge the forum also organizes a workshop at one of the premier computer vision conferences. So, conferences like international conference on computer vision or and pattern recognition CVPR or ICCV that is international conference and computer vision, so in such conferences this forum also organizes a workshop.

And the purpose of the workshop is to disseminate the new findings of the challenge, that is what are the new techniques or new models new methods that has been proposed and what is the success which are actually presented in these workshops. And, the purpose is the contestants they will come to know that and also the machine learning community, the computer vision community they come to know that what is the recent advances of the machine learning algorithms, right. And, the contestants with most successful and innovative techniques are invited to present their work in this workshop.

So, the next set of the architectures or the convolutional neural network models that we are going to discuss that I will present are actually winners or the successful entries in this ILSVRC or IMAGENET Large Scale Visual Recognition Challenge. So, with this I stop this lecture. From next lecture onwards we will talk about such new models or successful models.

Thank you.