# Modern Computer Vision

**Prof. A.N. Rajagopalan**

**Department of Electrical Engineering**

**IIT Madras**

## Lecture-40

Okay, so let's start. So the last class we did what is called the canny edge finder and you saw right and then how you can flag the pixels both weak and strong using what is called the linking this one mechanism. Now, the next obvious thing to ask is, can I get a solve a similar problem using my using what a deep network, right? And since you've done deep networks, right already, so the Oh, let me so, right, so we have, so we want to see right, how would one employ a deep network in order to solve this problem, right? So suppose I asked you, how would you do it? Suppose you wanted to solve the same problem using a deep network, right? I mean, you don't want to go through, well, not that you don't want to go through, but let's say that the alternative that I want to think about is given that the modern thing, you know, is to use a deep network. How would I go about? How would I approach it? What would I need? What would I need to solve it as a using a deep network? First of all, you would need an architecture and that should, that should maybe you know, need not be something very fancy, but you want something like kind of basic architecture. Will you go for an MLP or a CNN? CNN, because you have images, therefore, that part is kind of see decided. Would you need an RNN? Probably not, right? Because you just have one thread for which you want the, you know, edge of the pixels to be flat.

So there's no reason to go for anything else. So it's clear that I need a CNN architecture, right? Then what else do you need? And you would need actually a data set, right, which means and what should that contain, that particular data set? What should it have? What, what should be the constitution of that data set? What should it have? It should have images, of course, and it should have a ground truth edge map, edge pixels. I mean, otherwise, how will you train it? Right? If you, the simplest way to do training is to kind of think about some kind of a supervised approach. I will come later, you know, I will ask you guys about, you know, more sort of a difficult question later, but let us just look at the simplest.

The simplest is that I have an image and for every image that I have, right, I need a sort of an, you know, edge map. Now, who will actually give me the edge map? Can I use the, use the canny edge detector? No. No, because we are trying to find the substitute in canny. Exactly, right. So, if you say we are trying to try to get something which is even probably superior to canny.

If you give canny as a ground truth, then whatever you get can at best be canny, right, it cannot beat the canny edge detector. So, the issue is that you need a ground truth, you know, edge map which, which can come from, come from a human. So, that is why these human annotators are there. For example, you know, in a lab it could be students but then on a larger scale that it could be, it could be, you know, people, you know, to whom you pay to actually do this and people are willing to do this. So, what they will do is, you know, as a human would sort of observe, right, if I gave you an image.

So, you as a human would sort of think that, right, these are all the edge pixels which I think are actually, are actually the most important and should be caught by an algorithm. So, again, that is a little subjective. I mean, the same thing, give it another person that could be subjective variations, but it is okay. See, the whole thing about edges itself is that, you know, nothing is very concrete, right. The very strong guys are very clear, the very weak guys are very clear, the in between guys are never clear, right.

And therefore, there is going to be some, some uncertainty across people. So, if I gave her an image and asked her to draw an edge map, I do not think it will be identical, right. I mean, if I asked her, right, Aniruddha, it may not be identical, but that is okay. But then, right, on, in general, we feel that, you know, if you just, you know, give it enough number of people, those variations will get averaged out. And then the idea is, now you need, so you have sort of an input image and then you show it an edge map, right.

And then, and then you say that this network should sort of do whatever it takes, you know, learn the weights, have whatever it, different layers, have these convolution operations, maybe do max ruling, whatever you want to do in between. And then eventually I should, should kind of say, right, spit out, you know, a result that should be very close to the edge map, right, which I have, okay. This is how, this is how it should look. Now, one of the, one of the deep networks, right, that I thought I will just, you know, briefly mention is what is called holistically nested edge detection. This came in 2015 and then, then an improved version of this came out in 2019.

We will take a, you know, a brief look at both. So, for example, right, so if you kind of look at this, right, so here is, you know, it looks like a leopard or whatever. So, here, right, so here you have that. And if you really think about it, what are the, what are the most prominent edges in this image that you might want to catch and if you actually give it to a human annotator, probably, right, he will pick up something like that across people, okay, something like that is what, because you will just want to look at what are all the major things that I want to catch in this image, right. So, so the idea is that, right, now we are kind of, you know, going a little away now, right.

For example, we saw traditionally if you had to do it, right, then of course, you know, we would have had a certain way of doing it. But then now that, right, humans are involved, so we can actually use the strength of humans, right, in terms of what they would interpret as edges and so on and what could be important, you know, edges and so on. So, let us say, right, a human would probably annotate it, you know, this way and then you will have several such images for which you need the annotations, which is, which is kind of a laborious, right, it is not an easy thing because somebody has to sit down and do all this. But once you do it, it is there, right, anybody can actually use it. If it is publicly released, then anybody is free to use it.

Then if you kind of look at this HED is, HED stands for this is a holistic edge detection. So this, so this guy, right, so again this architecture, you know, is a little, is a little away from, from what you have seen in the sense that, right, it also has some side outputs. I mean, in the sense that it has losses at every stage, ok. Again, that is what, so as I said earlier also, right, deciding an architecture which architecture would work well for a problem, so people get those papers depending upon what they can show, right. I mean, if you can show that, you know, you have an architecture that solves a particular problem and there is a reason to have that architecture the way it is, that is fine, right.

So, so here these people argue that instead of, instead of, you know, just having one, one loss at the output, right, I mean, so, so, so what they are saying is instead of that you could have actually, you know, multiple losses at every, multiple losses in the sense that you keep showing the, the, you know, ground truth at every layer and, and then, and then you ask for the final output and then eventually I think when they take some sort of, when you see average, you know, a weighted average of all the outputs that kind of, you know, that, that emerge from each of the sides, like the side 3, side 4, whatever, then, then there is a final, so the final output is like a, is like a weighted average of what actually comes out of that. Of course, there are some, some, some small, small things, right, which if you read the paper it will be clear. The idea is not to sort of explain the whole network and all that. The idea is to sort of say that there is a traditional way to, the way for, you know, right, doing edge detection, which we have seen, right, which we understand now. Now, the question is if I had to solve the same problem given that it is a modern computer vision course, so if I had a deep network at my hands and if I had the knowledge of training networks and all that, can I now, you know, kind of recast this problem within the framework of actually, you know, deep network and solve it, right, that is the goal.

And you know, just a, just a glimpse of what would happen and, right, holistic in the sense that it is kind of end to end, ok, that is what the, the holistic, right, that is it being used in that sense. And, you know, a nested because, you know, at every layer, right, I mean, right,

you have an output and it is, you know, inspired by fully convolutional. So, all the way, all the way till the end, right, it is actually fully convolutional for this one classification problem. And it takes, you know, images input and, and the job is to directly produce the edge mappers output. So, and, and then some other things, improved speed and all that, that is fine.

And then the way this works is it is actually borrowed from VGGNet 16. So, it is again one of the standard architectures that we, that we, that we already saw. So, the architecture is the same, ok. So, they have taken the VGGNet and the VGGNet architecture, but what they have done is the last stage of VGGNet, VGGNet which had the red FC layers and all that, I mean, 2HC and how many. So, you had something like 4, 4 stages, right, and then after that, then you had a fifth stage, right, which was the, sorry, sixth stage maybe, right.

I mean, you have like what 2, 2, 3, 3, 3, right, somewhere 9, 13 + 3, 16, yeah. So, I think what you had was 2, 2, 3, 3, 3, right, that is, that was VHC, right, VGGNet and after that we have the FC layers. So, after the fifth pooling layer, so the one that comes after that which are all the FC layers, so including the, you know, fifth FC pooling layer, so, so they, so they say that they remove all of that and actually replace it, right. So, so kind of replace it with actually a convolutional layer and they say that is why it is called fully convolutional, I mean, so nowhere are you having FC layers and all, so it is completely convolutional. And so, what they are doing is after the last convolutional and then the side output layers are actually inserted after the last convolution layer in every stage.

So, for example, right, I mean, CON 1 which has actually, which has actually 2, 2, 2 layer, I mean, right, 2 layers, right. So, they are saying that after the second layer, similarly CON 2 after the second layer, then CON 3 because you have 3 layers, right, therefore, after the third one and similarly CON 4, CON 5, they all have 3, 3, right. Therefore, so, so what they are saying is after the kind of third feature map, right, so that is where they actually insert a side output layer and they of course make sure that the, that they say dimensions and all match, I mean, so for example, this dimension because finally, right, if they want to take a weighted average, then you need similar dimensions, right, so that and all is taken care of. They will have max pool and all which then you will think that well, we will not be, kind of say dimension go down, but then there is something called a deconvolve operation which I did not talk about, right. There is something called just as you have a convolve, right.

There is also what is called a deconvolve operation. Actually, that is not strictly speaking, you know, in actually traditional, you know, signal processing, you know, they say decomposition actually means some kind of an inverse filtering, ok. That is what decomposition strictly means, but in deep network literature, it is kind of abused, ok. Now,

this simply means that, means that, you know, I mean, if you have reduced a dimension, right, then you want to get a, you know, go back to the original dimension, but in a kind of a systematic way, ok. So, that, that is, that, right, I did not talk about, you can actually read about it, it is nothing, it is nothing very, very fancy, it is a straightforward to understand.

So, of course, you know, you need that, you know, deconvolve layer because you are kind of shrinking the dimension, but then, then you want to finally, right, do some kind of a weighted average and therefore, they all should match in terms of, in terms of the size. So, but what they are showing is at the end of, end of every, every stage, right, they are showing the, showing the output map, the output edge map that is needed and there is a loss there. Again, at the next layer, show the output map, there is a loss there. I mean, these are all things that you can ask them, you know, why did they do at every layer, could they not have done it, you know, the third layer, but those are what are called ablation studies. So, so, that paper they would have done all that.

They would have shown that, you know, if you kind of do it otherwise, then, you know, then basically it does not work that well. So, all those are part of what is called, you know, anytime you write a paper, there is something called an ablation that you have to do, which basically means that you have to prove, you know, why you have these things, right, which you have there. You cannot simply say, you know, I, I like to have them, therefore, they are there, right, nobody accepts that. So, you have to convince, right, we are the other, you know, convince, you know, whoever is reviewing, you have to convince them that all of this makes it, which, which I am sure, right, that is all there in the paper. So, that is all the ablation studies and then the final, right, HED output is a weighted average of all the inside outputs and of course, there is also a final output of this, you know, this one, this one, a network and they are computed only over the final output of this network, but on all the inside outputs, right, as I said.

So, now, here is something, right, so that, that you can see. So, what they, what they mean to say is, right, so, so I think, ok, so what they mean is without, so, build deep supervision means that every layer, right, if you, if you had this kind of a, kind of a supervision, then, right, I mean, you know, according to them what you actually end up, end up, end up in seeing, right, this is actually much superior, they are not, they are not really having, having this kind of, you know, a deep supervision, ok. So, really without means that, you know, at those side outputs, if you do not use that loss, then what happens versus using it, ok. So, again, right, these are all studies, ok, which, which, you know, which they have, which, you know, which actually established that. So, in the, in the, in the first example, it is some kind of a bear, right and you, and you see that the, the, I mean, final output, right, looks, looks, looks actually good and similarly, right, similarly here is actually a building

example and I think, you know, I have some more examples there, but then the point is, so, so the point is this, right.

So, you can actually solve the same problem. Of course, you know, if you tried Canny, right, so Canny output is also given here, I think somewhere, yeah. So, for example, if you have tried Canny, right, with let us say various sigma, so there is sigma equal to 2, there is sigma equal to 4, there is sigma equal to 8, you get something, right, like that and of course, you know, if you had a reduced sigma, then you will have, of course, more H pixels and if you had even less, then of course, you would have even more H pixels and so on. So, if you try to compare this with this, right, probably this is not as great as what you have, for example, the output, right, that is, that is emerging out of the network and for obvious reasons, right, because you are, I mean, right, you are actually doing, you are sort of, you know, giving the ground truth, you are showing what it should look like and at the end of the day, right, that is where I think deep networks come up with this power that, that, you know, once you train them well and once you train them with a rich kind of, see, sort of a data set, then that, then when they kind of generalize outside, outside in the sense, they are not too far away, but then as long as, there you do not have to show them the same example, but they can still do, you know, a decent job and it is very fast and all that and therefore, right, these days, you know, you have kind of a deep networks that can actually do this H detection job. But I think, but the point is we should not forget the fact that we should also know where they come from, where they just come from.

See, see, see, the problem with just doing a deep network is that if I, if I, if we had just learnt about how to do this, right, you would not even know what is an edge, right. Suppose, let us say, suppose I had skipped all of that, right, suppose, suppose, right, we had not talked about edge orientation, edge magnitude, nothing we had talked about and we said, okay, if you want an edge map, right, throw this, throw this ground truth, right, annotate and show you will get an output, right, you will be able to, you know, maybe even arrive at a kind of network that does the job, but then your insights are rather weak because you do not even understand what is an edge pixel, right. That is the reason why we always straddle between both, right, a traditional thing where we actually, where there is a firm grounding in terms of, because that is what was there all these years and then, and then using that knowledge if you can do something, okay, in this case we are not really using any of that knowledge per se, but if you can, right, that would be good, okay. That is what I call like inspired, what do you call, what is that, what was it, what inspired, what inspired, physics inspired, right, physics inspired deep learning, right, that is like things that you have learned from a traditional approach and then if possible use it. Let me ask you a question, can this be done, I mean, right, I mean, if you ask me whether I have the answer, I do not have the answer either, okay, but let me ask, right, can this be done without a ground truth, what is called self-supervised learning, right, that is where the, that is where

the area is headed, how do you do an edge detection, how do I arrive at this output without me showing a ground truth, can it self-learn to produce a depth map, not depth map, edge map, okay.

Then further, you know, improvement on this came in, you know, in 2019, right, which they called richer convolutional features for edge detection, actually it turns out that, you know, I do not know whether it is the same bunch of authors, but roughly the, you know, network architecture everything is the same in the sense that, in the sense that, right, I mean, you know, you have, you have a similar thing, but the only change that you see is that, see for example, so if you start from here, right, you know, inputting the image from there and then you have, this is again a VGG net 16, so you have, right, I mean, 2 conv layers, 2 conv layers, then you have 3, 3, 3, 3 and then the final, of course, you know, the final fully connected layer is all, is all, is all removed, but what you find is there, right, in the earlier one, I mean, in the, in the earlier one, right, so the feature loss was being calculated here, whereas here what is being done is, you know, both these channels are being used, right, so you see that, right, you know, so you see that this output is also being drawn from here, this output is also being drawn from here, then there is a 1 cross, 1 cross 1 convolution, okay, right, being applied on both and those are actually summed up and then again a 1 cross 1 convolution and there is a loss here, similarly, right, there is a loss, so that part is still the same, but then in between, right, they have shown that if you take information from, from, you see, every layer at that, you know, in every stage of the, let us say, VGG CNN, CNN blocks, then they say that and then of course combine them, you know, in a suitable manner, so anyway 1 cross 1 convolution and all we saw earlier, right, so it is a kind of very, very useful to, to be able to, you know, match, you know, this one, dimensions and so on and therefore it, okay, that they do and then finally, right, what they do is they actually fuse, so the fusion is actually a concatenation here, so they simply concatenate there, it was a weighted average, here they concatenate and then followed up with some 1 cross 1 convolution and then of course, you know, then there is the final, then there is also the output loss and if you do it this way, it turns out that, that actually, so the L twice means, I think, right, element wise layer, okay, so if you, so do not wonder what that L twice means, I think it is just element wise, so here, for example, they are summing up, right, it should be element wise summation, therefore, what they mean is, so, right, you are able to do that and then I think, see, see this, see this guy, that is a deconvolved block, okay, that is the one that makes sure that you are, that you are, see dimension, dimensions match up, so that you can concatenate and so on. And if you do this, then they show that, for example, the top line is the original images, from 2nd to 6th line is the output of stage 1, 2, 3, 4 and 5 respectively, so you see that, so this car has gone in and then, right, this is what has emerged as the output, okay, so right, this is the final output in a sense and here, right, you have, so, so what you see is, you know, there are finer, finer edges in the initial layers and they kind of probably combine you know, in a sense to evolve into more,

more, see meaningful edges, then they evolve into more and more meaningful edges, eventually leaving view with edges that kind of seem to be the most appropriate for the task, but this is all coming from a ground truth anyway, right, so you are showing a ground truth edge map. And they also show that, show this kind of multi-scale algorithm, so by which what they mean is, what they actually mean is that, that for example, right, I mean, so if you actually, if you actually gave this, gave this image, right, at the, at the right, you know, whatever, right, this one, this one, right, original size and for which, for which right you have an output, then if you actually scale it down, just as, just as I talked about the, you know, Gaussian scale space, so you actually, so actually blur it, then you see scale it down and then, and then, and then, and then, right, you try to find out the output again, again pass it through the same network, then you get an output which is actually, right, you know, reduced in size, but then the idea, the idea being that, being that, you know, because, because of the blur, right, you would have probably masked out some of the unwanted edges and so on. And then if you again interpolate and bring it back to the original size of what you were getting from the actual size of the image, then you can go further down, do this and then again you kind of bilinearly interpolate, okay, so it is like a four neighborhood, right, that is what we mean by, just as in the other problem we had linear interpolation, right, where we had only two pixels as neighbors because when we were doing a gradient problem, here it is like intensive, it is, it is like four neighbors you will have and then you can do a bilinear interpolation and then bring it back, right, I mean, again, right, okay, depending upon the size by which you are going down. And then if you just combine them, right, so and then, so this is that, this is what they call as a multi-scale thing, with the same network, right, they are not changing anything.

One, one way is to simply use that network, push this image and then this is what you get. Another way is you push the image, this is what you get, then you, then you kind of push a, push a down sub-sampled image, get this output, push another sub-sampled image and then you show that, you know, the output of this, this is what you get and then if you compare this with this, right, this is what you would have got with just one image and then if you use a scale, scale sort of, you know, pyramid, then this is what you get and their claim is this is slightly better, but again, I mean, you know, you can, you can argue, you may probably say that this is a little more sharper, maybe, maybe, right, you should actually accept that, but again, right, this is some paper that they have done and some analysis that they have done, if you take the weight, say, simply the average, not weighted, I think, just, just average them, then you get something like that and they say that, you know, this is, maybe the claim is that this is actually better or whatever, okay. So, and all of this, so the, so the, so the key thing being that, that even, right, as, you know, as early, I mean, you know, as early as 2019, right, it is not like, what, you know, by our standards it is not that far away, so you can see, still see that, right, even though you have deep networks and all, they still need a lot of supervision, right, so the, the real challenge is to

sort of, you know, do it in a self-learning form, right, that is where the real challenge is, that is where I think, I think, you know, your own input should come in, you know, where you think, you know, in a certain way, where all your, all your, you know, all that you learn, right, in a sort of a traditional sense, can that be, can that be, you know, brought to, brought to some kind of a bearing, right, into this problem, inject that into this, whatever, whichever way you want to do it.