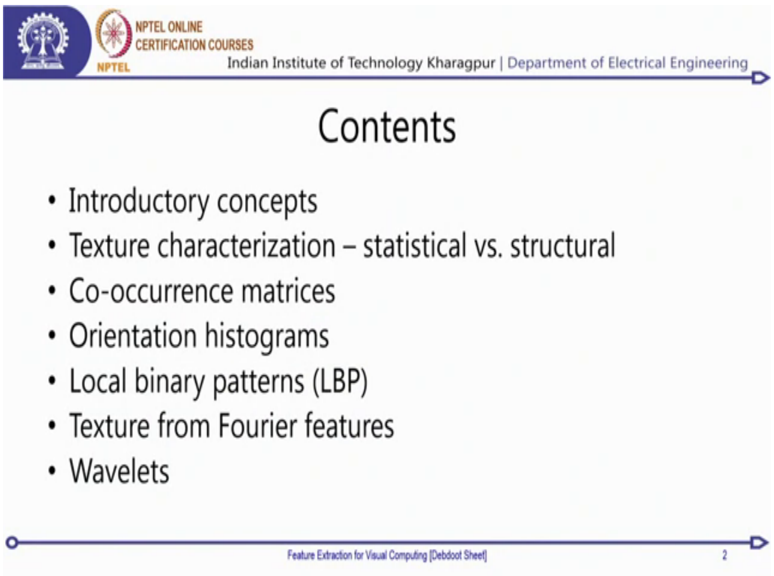
Deep Learning for Visual Computing
Prof. Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 02
Feature Extraction for Visual Computing

So, welcome all. In the last lecture we had learned about the whole aspect of visual computing giving you an introduction on what it is, and aspects of a deep learning is going to impact. So, before we really jump start into it one thing which I had said was we need to understand about some of the classical aspects, and one of this is how was classical visual computing done using the representative aspects of features, and what where features and how this feature extraction was classically carried out otherwise in in our classical terms. So, this whole lecture is what is going to focused on feature extractions for visual computing and we would be going through it.

(Refer Slide Time: 00:59)



So, without waiting for much, I would start down with an introductory concept explaining you what visual features are and what are textures, which are some of the major visual features which are taken down into consideration. As far as for it is closeness to understanding deep neural networks and how they work out, we would have to understand down much richness about these neighborhoods of pixels and what is locality of a feature and when different pixels are in the neighborhoods and long range or short range neighbors, then how do their relative differences and their intensity and appearance really impact out the total model of inferring about an image.

So, from there we will start down initially by understanding the differences between statistical and structural textures, and enter down in to very classical methods called as co occurrence matrices then go down to orientation histograms and some of the recent rather recently developed contemporary techniques called as local binary patterns, then enter in to textures from Fourier transforms and Fourier features and then go down into wavelets which are which is as such the family which is closest down to a deep neural networks or very specifically convolutional neural networks. In order to understand down, how features are learnt down and how they relate down to different aspects of how we see images.

(Refer Slide Time: 02:20)

So, if you really look into an image over there. So, what do you see over here; is basically a car driving on a speedway, and it is a very scenic environment over here if you look into the image. So, there is a mountain there are some trees over there and a clear blue sky like we have quite a interesting day to take a ride actually a really nice day to go out on an outing, but then sad part is that we will have to do this lectures, and the interesting part is that you are somehow interested in understanding, what deep learning dues in does in understanding these kind of interested ones.

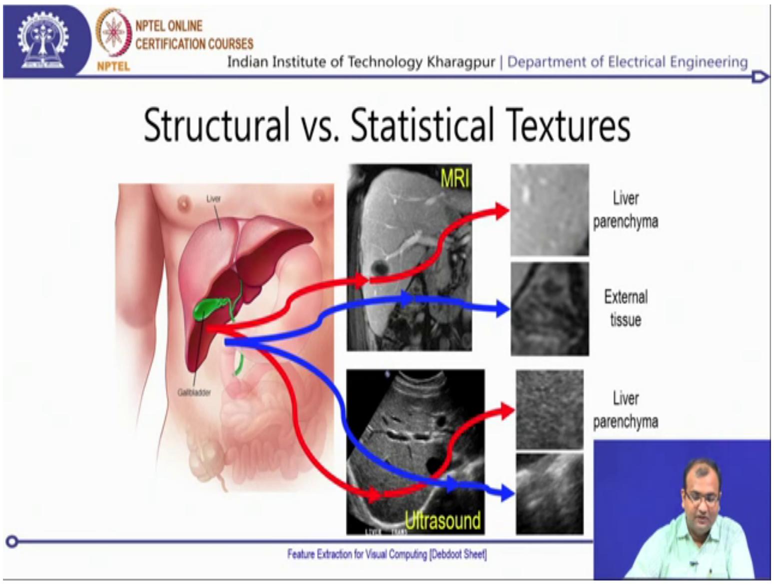
So, let us go into the classical part of it now if you look into this whole image over there, you can see the different parts have different kind of aspects. So, you can look into trees and this is what the sort of tree appearance looks like not really classical part of a tree, but just a small region around tree and that would have some part of the greeneries come down over here, and a part of the background mountain coming down.

If you look at a part of the road then definitely it is black there is amount of tarmac, there is a good amount of shadowing from the and from underneath the hood of the car, and then apart of the front protection gear of the car is what comes down there is a part of the sunny streak on the road as well. It is its quite heterogeneous and that is the one which I wanted to really look into it now. As we go into it, you have 2 different aspects one of them is called as textures, and that is what is called as a local variation in intensity due to some sort of an heterogeneity or. So, this is just says down as tissue heterogeneity, but it is its some sort of a just a local heterogeneity which you see over there.

Now, on the other side of it you have noise, and that noise is something which is due to the uncertainty of those sensing instrument itself. And the noise over here is that we know that there are just trees and leaves over here, but then you would see certain points which are darker than other certain points, which are just brighter than others and this is sort of just a disruption cause down by the image sensor itself. So that, that is what happens down in terms of images.

So, you have textures which are local variations, but you also see down noise which do manifest themselves as local variations, but are more of actually something which is related just to the sensor, and you have a major role to play neither does the image formation process form any significant role. So, similarly there is another aspect which is called a structural versus statistical texture, and the way we craft down certain of our future descriptors is what is heavily guided down by whether it is what we are looking down a structural or what we are looking down is of a statistically varying nature.

(Refer Slide Time: 05:01)



So when we go down through this one. So, let us take into the internal anatomy. So, my favorite is medical image analysis. So, I would be pulling down a lot of examples from there. And what comes down is say I am looking into this part of it which is my liver and then if I try to look into the MRI magnetic resonance image of my liver over here and let us draw correlation. So, this is a point which would map down to this exact point over here. And now I can also try to look down into an ultrasound image of my liver, and that is by using acoustic waves that it takes down those images right.

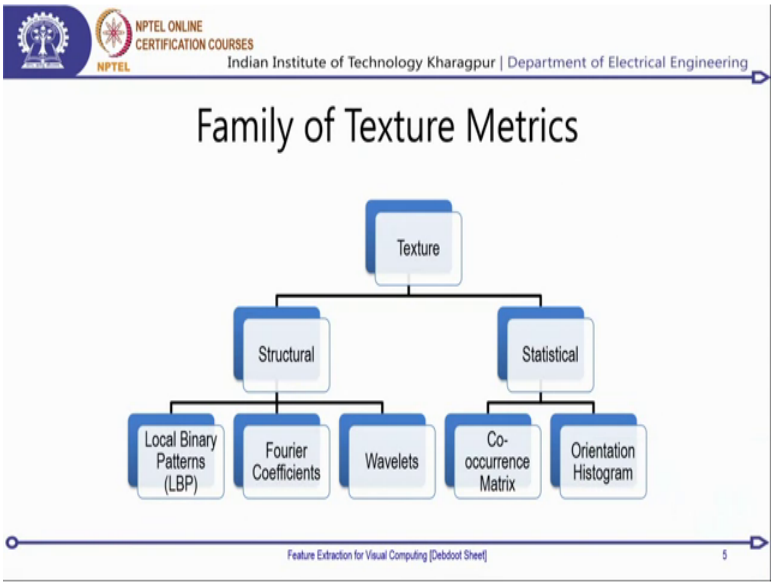
And if you clearly look into it the only, similarity possibly you would see is that they are all grayscale images, speckles are brighter than the background which is just dark over there and this is how these are from. So, and an other than just it being grayscale image there is not much of a significance, and the only other commonality would be that the organ being imaged over here is just a liver.

But then let us look into how you would look into different aspects of the textures over there. If you look into MRI and within a part over here which is called as a liver parenchyma or the mass of the liver tissue over there, this is what it looks like if you look into a part of the liver parenchyma within your ultrasound, then this is what it looks like. A bit noisy over here quite a bit smooth and steady features, but here it is like really something which is not so, soothing to the eye to look into it.

On the other side of it if I look into a background region which is a region outside, this sort of an anatomical region and then let us see what it looks like. So, on the MRI it will be showing down some streaks and definitive patterns, and there is definitely some sort of correlation which you can draw down over here. On the ultrasound part over there the interesting part is that this also looks the same kind of a noisy and one and then it is really hard to draw down any definitive correlation. So, this is what guides down our understanding of statistical and structural features of textures.

So, what happens on the earlier case over here as in MRI, you would see that they are sort of stationary and there are distinct structures which you can see down in the background as versus the liver parenchyma. Whereas, if you look into over here then the major interesting fact which comes down is that it is a sort of statistical in nature. So, all though the I mean there is not a distinct structure which you would be seeing, but what you definitely see is a lot of relative change in this speckle intensities, which come down over here. And that is what guides down our statistical versus structural matrix to be used on for describing features.

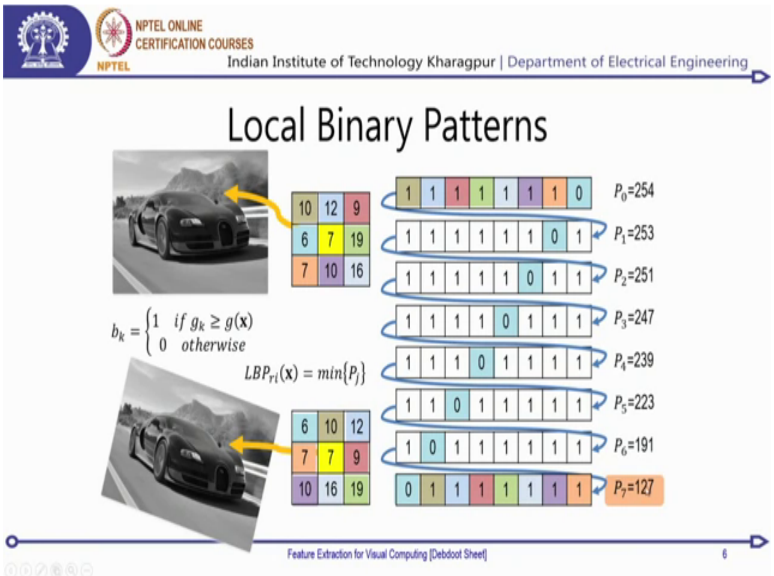
(Refer Slide Time: 07:40)



Now, as we go into the different kind of family of these texture descriptors, it can broadly be divided into 2 broad zones and they are on statistical and structural. Now on the structural part of it what we use for identify structural features are wavelets, your coefficients and local binary patterns.

On the other side of it which consists of statistical features we use orientation histograms and coconuts matrices. Now we would enter you into all of them one by one and go through it to have an understanding of what these features actually mean and how they really impact us. So, let us start with the first one which is called as local binary pattern.

(Refer Slide Time: 08:24)



Now how it starts with us say that I have an image and I take a small region over there and now this image has to be. So, for local binary patterns to be calculated you need grayscale images, and for majority of the ones which we are calculating it is its grayscale images which you would be taking down. Now you can definitely use color images there are different ways of doing either you can employ the texture descriptors independently on each of the channels over there, on the red channel, green channel, and the blue channel the other way maybe you transform the colour image onto a grayscale image and do it you can transform it into any other color space and do it or for certain of them you do definitely have a descriptors which are a vector valued.

So, you can take in a color vector over there and then start computing on top of it. But for keeping things simple we would be starting on with the simple local binary pattern as of over here. So, what it starts with is say that I look into one particular region and let us take a 3 cross 3 patch. A small patch from where this arrow is pointing over there. So, the centroid of the pixel which I am looking down is the one which is colored in yellow over here and this is sort of how the rest of the 8 neighbors around that particular pixel is arranged.

Now what we define is we define a small code sequence which is called as bk to be 0 or 1 based on this one. So, k is basically one of these indices over here. So, this can this is k equal to1 k equal to 2 k equal to 3 k equal to 4 and accordingly. So, you can vary your k like any ways, and you can go clockwise anti clockwise and, but you need to give the same convention while applying through all the pixels on the same way of h now the idea is that if the neighbor over here, has a value which is greater than the center value, then you replace it by a 1, and if it has a value which is lesser than that then you would just replace it by a 0. So, that is the only way how a binary code sequence over here for LBP is defined.

Now if you look over here, this is how the sequence would get created if I start from this particular pattern and that maps on to over here, and then accordingly go down in a clockwise fashion and this is what it generates. Now interesting and then not so hard to generate as such anybody can generate just a binary code. Another interesting attribute is that since there are 8 neighbors around this 3 cross 3 patch over here. So, you will get down an 8 bit sequence right. So, it is just a 8 bit number which you have over here it is an integer 8 bit number which comes down over here. The interesting fact comes down that say my image just rotates slightly.

Now if there is a slight rotation on my image then I get my image like this and then if I take down the neighbors the neighbors would look something like this well then my and then let us start down computing out what is the LBP. So, I would be starting down at this point we just make a equal to one in anyways. Now what happens is; that as you look through this kind of a pattern being computed out over here, what really turns out interesting is that there has been some sort of a shift, and on view of this shift what comes down is that you get a different 8 bit number.

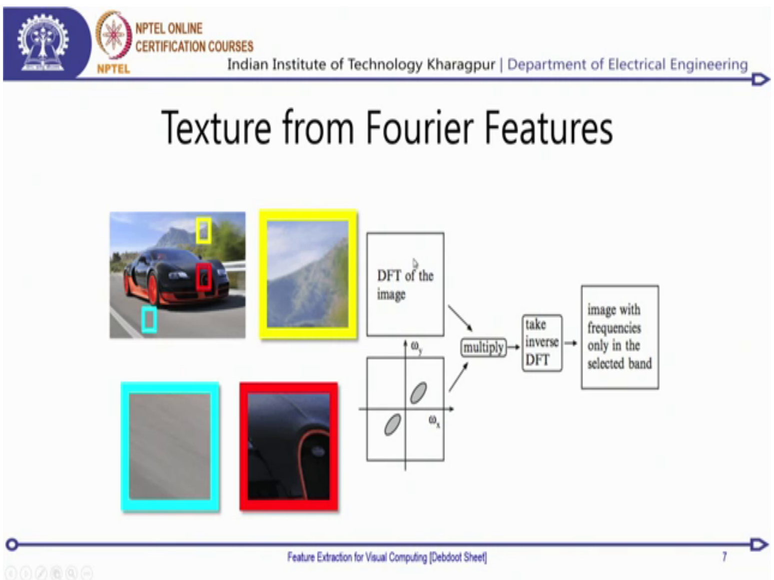
So, in the earlier case you had a 8 bit number which whose value was in the integer space 254. Now you got an 8 bit number whose value in the integer space is 127. The point is that my image is the same my pixel location in both the images is also the same. So, apparently my descriptors are also supposed to remain the same, otherwise if I just tilt an image you would make a different inference out of it, that that is something which is like really awkward and really critical in a major way. So, what we do is in that case we can find out a trick of solving it out.

So, let us get down an arbitrary pattern with one of them, and then we keep on rotating it in a circular fashion. So, you remember your circular shifts operations from your DSP classes, now we do the same kind of a circular shift over here. So, as we do the circular shift we get a different 8 bit number every time coming down. So, for each 8 bit number you have an integer representation over here. Now LBP is typically defined as the value of the minimum of these rotations.

So, typically if I have an 8 bit number, I will be able to do 8 times of those rotations and get down a sequence. So, there will be a sequence P 0 and P 1, P 2 and accordingly going down till P 7. Of each of these sequences I will find out which has the minimum value. So, over here it incidentally turns out that P 7 has the minimum value, and whichever is the minimum value that is the one which assigns has the LBP at that location. So, what it necessarily ensures is that you can have the image rotated and still there is no problem you can finish it off, and then get down a minimum value coming down.

And every time you have any arbitrary rotation, it will still give you the same value for the LBP coming down. So, this is one of the descriptors used on for understanding images in the classical context.

(Refer Slide Time: 13:14)

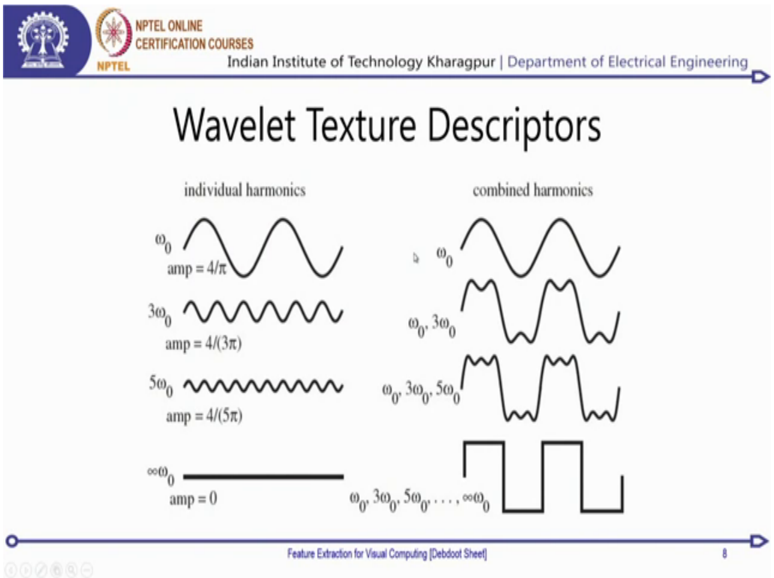


So, the other one is where we make use of Fourier features in order to understand them. So, again getting back to the old example, say we look into a small region around the mountain and the trees then we look into a small region around the hood of the car and then we look into a small region on the road. So, if you clearly look into these 3 regions they do have definitely features. So, this this road part is sort of flat and had directional streaks over here this is sort of land which has. So, a slight variation coming down and a streak of red coming down, and here there is a sharp boundary which happens between the mountain and the sky over there. Now each of them will be giving down a different part of Fourier descriptor. So, if you take a DFT of the whole image patch over here, you do you know different sort of topological values and let us say that the peaks are located somewhere over here.

The idea is that if you know what peaks actually correspond to what sort of regions within the image, then what you can do is you can take the DFT of the whole image and then multiply it with this sort of a filter over here, on the Fourier space and then you take an inverse DFT. So, what do you get down is you get an image which has only those bands of frequencies prism which correspond to a certain texture or which will correspond to a certain image. So, now, I will be able to find out only that image wherever. So, that that kind of an object wherever it look is located on the image, now comes to me very easily weather without much of a problem by just using these Fourier descriptors itself.

So, this is another classical way of doing it and as we enter down into our neural networks, and then eventually onto convolution neural networks and how they are implemented, we would be getting down into a much deeper and detailed understanding of how in fact, this whole theory of Fourier’s does help us in realizing our convolutional neural networks, and why they make much of an affective sense as well.

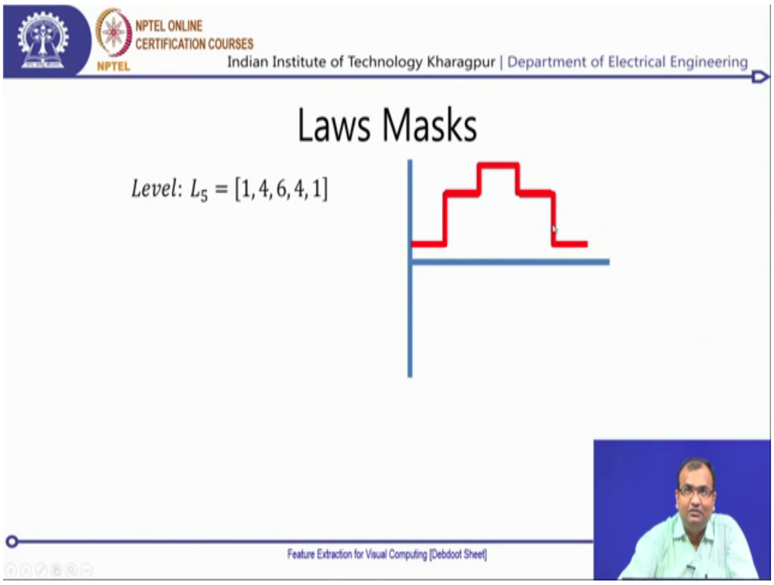
(Refer Slide Time: 15:13)



So, from there we get down into a classically what is called as wavelet based texture descriptors now. So, the theory is not so complicated. So, what happens is on the left hand side what you see over here is basically some sinusoids and each of them is a harmonic multiple of the previous one or the fundamental plus the harmonics over there. Now you remember from your Fourier series theorem that as you keep on adding down weighted combination of these harmonics on to your fundamental, you would be approaching down quite to a kilo this kind of a square wave, and that is what is being or what is one of the classical first examples of your understanding of Fourier series itself.

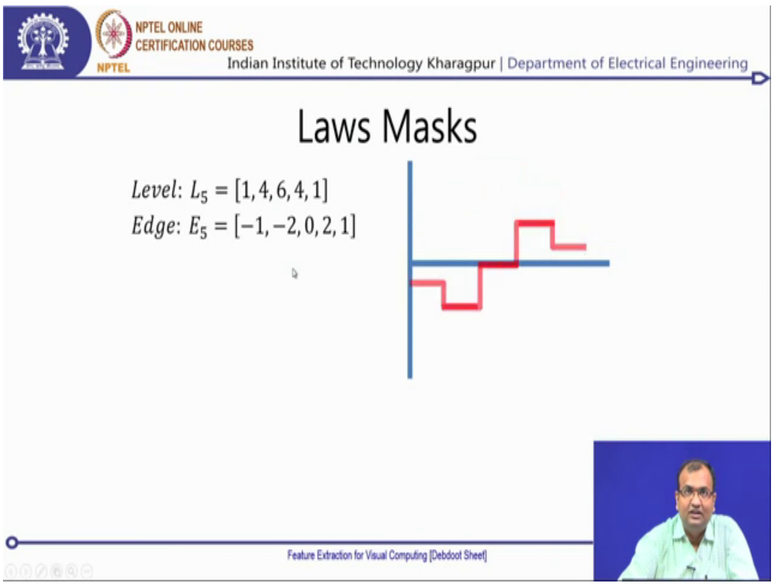
So, we make the other way round over there now say that if we have an image which can be decomposed into small sort of waves and called as wavelets, and then they can be used in order to find out. So, here what would come down is that here you see that there is a coefficient being added down over there for each of them. So, we can tell them what is the weight and just told them what is the weight over there to go down. So, let us get into a very simple one which is called as a loss mask made out of integer valued wavelets, which will help you in understanding what wavelets actually are.

(Refer Slide Time: 16:32)



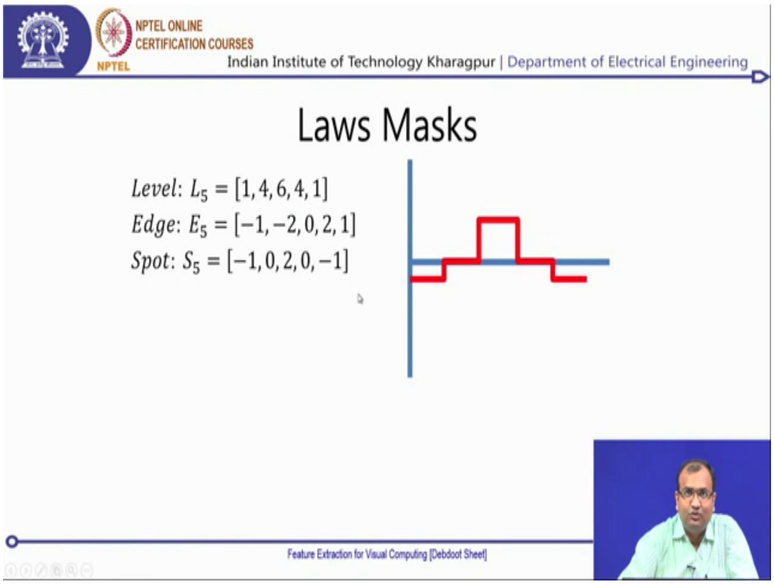
So let us look into the first one over here. So, we are looking into a series 5 wavelet which has a length of 5 elements over there and what we do over here is a level wavelet. So, the values over here are given something of this sort, and the profile is some whatever here. Now if you carefully inspect this this almost approximates a Gaussian like behavior over there which peeks down at the center point which is 3 and as you can convert, you will be getting on a different aspect coming. So, we will come down to why it is called as a level over there.

(Refer Slide Time: 17:07)



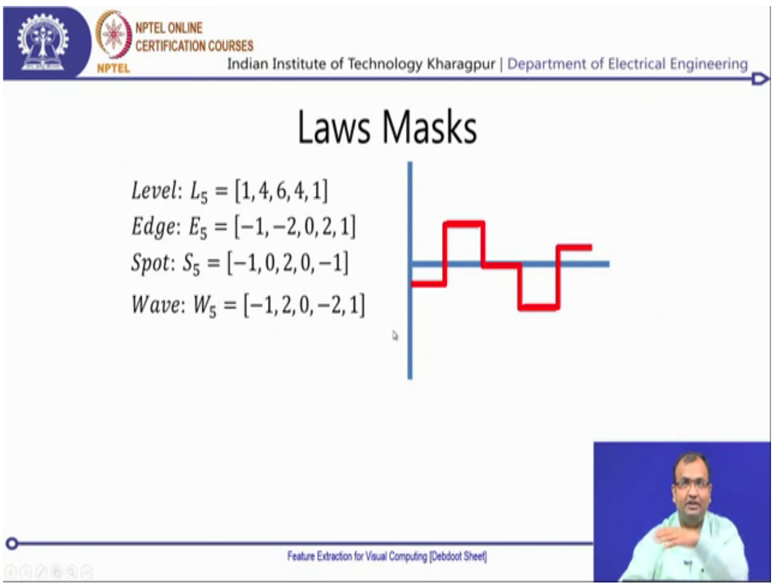
The next one is called as an edge wavelet, and which is 0 some wavelet the earlier one is a non-zero some, if you sum down all the elements. So, typically if you run it over a flat region you will not get a 0 output over there whereas, for this one if you run it over a flat region you will be getting up your 0 output and.

(Refer Slide Time: 17:26)



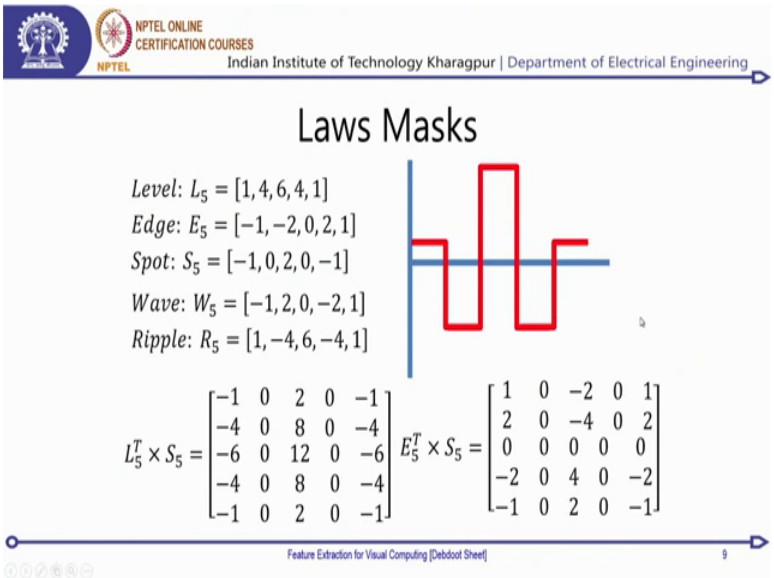
The next one is what is called a spot and these are what detect down spots present over there, and if you carefully inspect this is also something which looks like a Gaussian, but then a shifted version of the Gaussian itself.

(Refer Slide Time: 17:37)



The next one is a wave, it has sort of a very nature that is why it is called as a wave and this called as a ripple because of it is rippling nature.

(Refer Slide Time: 17:43)

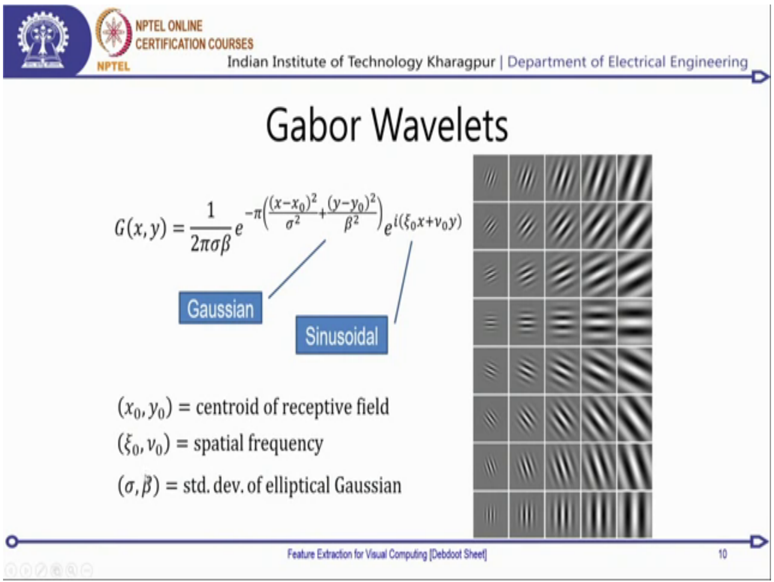


This rippling nature does replicate quite close to a invert of a Mexican hat or invert of Laplacian of Gaussian itself. Now what we do typically is that we can take a cross product between the transpose of one of these and the other one in order to get down different sort of 2 d wavelets coming down. So, like this is a cross product between level and this spot, and this is the 2 d wavelet which we get down coming together over here.

Similarly I can take down a cross product between the edge and the spot and this would give me different kind of wavelets. Now what we typically do is that you can use these sort of corners as convolution corners, and then you can convolved over the whole image in order to extract out to your features, and these features will be pole pixel basis features which you can now use for subsequent analysis itself. So, these are the different sort of ways in which you can use it for your computing in a classical way. Now as we get down into our c n ns, we would be realizing that some of them would be converging onto giving you kernels like this and these corners will be data driven, they are not given down by some inventor who had to spend a lot of time.

Solving out mathematics of understanding images and then coming out with these, but then it would be simply directly given out by the data itself, and we will learn down how it actually happens.

(Refer Slide Time: 19:14)



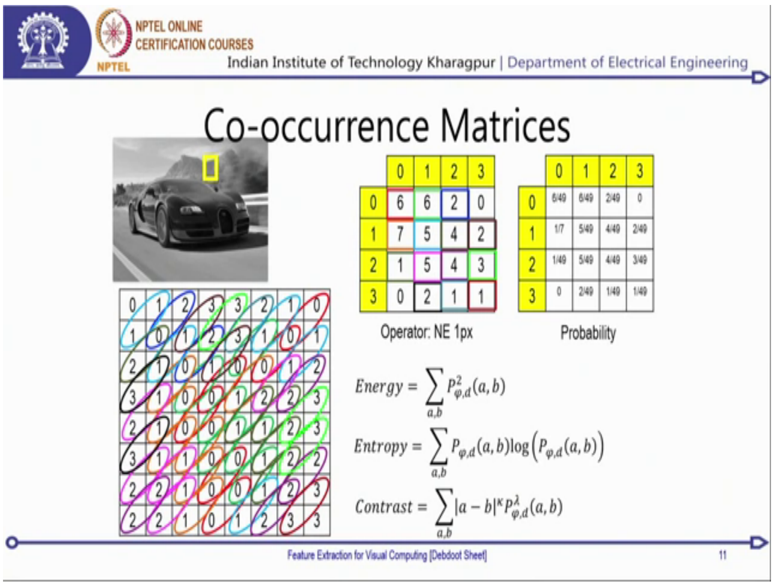
Another classical descriptor is called as a Gabor wavelet, and these are very famous for their contributions to understanding biometrics. So, understanding your finger print patterns and your iris patterns in order to really isolate and then provide you and validate your identities. So, this, whole wavelet is made out of 2 parts, one of them is the Gaussian part if you look over here, and the other part is the sinusoidal because you see this complex imaginary term over here which makes it sinusoid a complex valued term. So, you have a repeating sinusoid coming down.

Now over here there are 2 aspects, one is this x naught and y naught which is called as the centroid of the receptive field where exactly your wavelet kernel is located zeta naught and nu naught are the spatial frequency. So, these are the x direction and y direction spatial frequencies for your sinusoid which comes down and sigma and beta are basically standard deviation of the elliptical Gaussian over here now it looks like a complicated equation, but what for turns out there is quite interesting.

So, if you keep on varying these values of zeta naught and nu naught and sigma and beta for a given x naught and y naught, and then you keep on changing the values of x and y as in over here then these are the different patterns it would be generating on the 2 d space. Then if you carefully look into them these also do appear as small sort of wavelets and then what you can say is that if you go along all the columns in a row, then you would see that your frequency is changing your. So, over here your y direction of frequency quite is changing over here whereas, if you fix down yourself along one of these columns, but traverse across all the rows then what you would be seeing down is that the direction is changing over there.

So, this is something which f gets affected in terms of the direction change is something which happens in terms of sigma and beta coming down over there and as well as your zeta naught and nu naught. So, if your zeta naught is high whereas, nu naught is 0 then you would be getting this one whereas, if your nu naught is high whereas, zeta naught is 0, then you would be getting this vertical standing thing and all the intermediate ones lead down to this one. So, just by changing down you get down some sort of a circularly rotating out wavelets as well for your future descriptors.

(Refer Slide Time: 21:33)



From there comes down co occurrence matrices and another interesting one. So, what happens with the co occurrence matrix is quite interesting. So, if I take a small region around over here. So, this is what this whole small region might look like. So, this has 1 2 3 4 5 6 7 8. So, it is an and on this side also you will be seeing down, that there are 8 rows in one single column.

So, it is a 8 cross 8 region which I am taking down on this original image, and now what I would like to do is that I define some sort of an operator which is called as a northeast one pixel or basically it is something which is looking in the northeast direction this direction and it is looking at one pixel levels. Now what I want to look into is that I want to see how many times 0 is co-located with a zero. So, let us count down. So, we have this kind of relations where if I stand on one particular location which is 0 valued, and I look into it is north east at one pixel then I see that there is another 0.

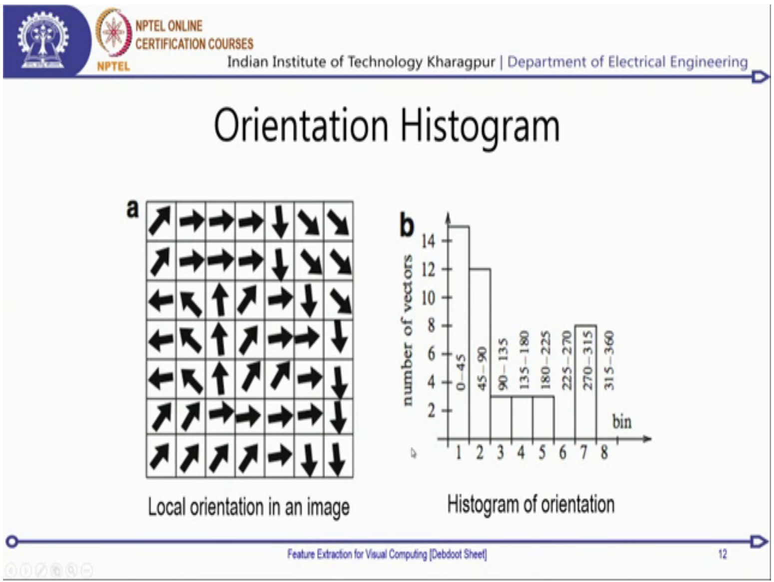
So, what we have is typically then there are 6 times that this kind of a occurrence comes down over here. If we try to look into the other aspect over there which is a little standard a 0 and look into 1, which is standard a 0 look into one I get six of them. And similarly so on and so forth what we can do is stand at a 0 and try to look into the value of 2, standard a 0 and try to look into the value of 3, and we keep on accordingly going down.

So, if you standard a 0 and trying to look into the value of 3 you see that there are none and that is why this value is 0. So, accordingly we keep on calculating over all the pairs which are being formed over there and then we can keep on populating this matrix which is called as a co-occurrence matrix. Now as the name suggests that 2 values need to occur in pairs or occur together and that is why this code term comes down over here. Now that this co occurrence matrix is comes down what you can see is that there comes down a co occurrence probability as well and this is 6 by 49 this is 6 by 49 this is 2 by 49.

Now thus, this denominator over here is basically sum of all the possible elements which comes down over here, all the possible co occurrences which can happen and that is also related down to the size of this small patch which we are taking down. So, say I took a 8 cross 8 patch, and I am looking at a northeast director. So, what will happen; is that I can start from this point only and then try to look into it, and as I go on this side my last point of origin would be over here I mean I can look down. So, technically I can just scan through 7 columns and I can scan through only 7 rows in order to get you down.

And that is why the total number of combinations over there would be 49 7 into 7 and nothing beyond it. From there what we can find out from this particular probability matrix over here, I can find out what is my energy of co-occurrence, I can find out my entropy of co-occurrence, I can find out something defined as the contrast from my co occurrence matrix and these are the features which are used on subsequently for describing a small part of an image. Now given all of this the second the next one which comes down is an orientation histogram.

(Refer Slide Time: 24:34)

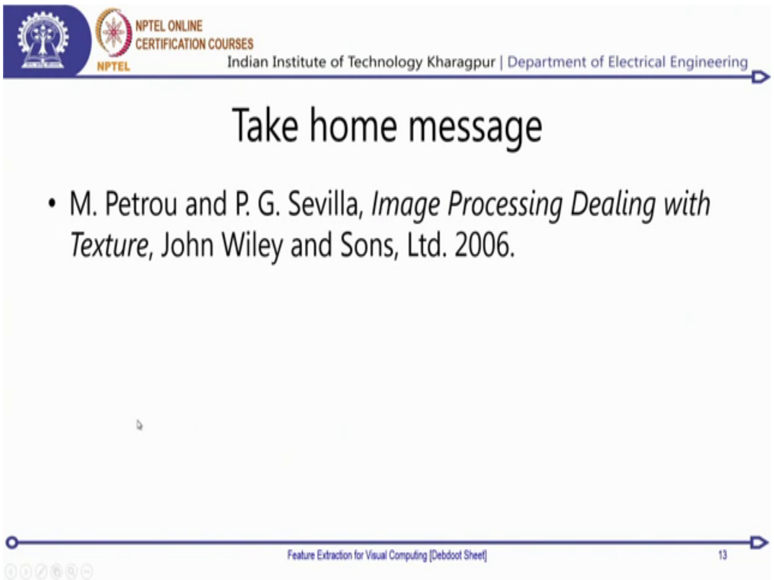


So, whatever intention histogram typically does is that if you have an image you run down a gradient operator. So, one is you get down your amplitude of the gradient the other part is that you get down the direction of the gradient. Now if you look into the direction of the gradient and try to quantify them in to different angles say over here I am quantifying it into 7 different bins of different angles, and then I can look into how many such blocks or how many such locations within the image have a gradient which is pointing along this direction.

So, this is on my direction of 0 to 45 degree, how many are pointing in my? 45 to 90 degree, how many are pointing in my 90 to 135 degree. So, every 45 degree if I am just bringing it out. So, I will be having total number of 8 bins, which have across which I can put down my directions of my gradient and this is another form, which is used on for calculating out the local features within an image itself. Now having said all of this these are some of the very classical ones which we will be using.

So, in the next class where we would be doing a lab exercise, we would be looking into how these features are calculated from an image and what are the different software tools you can use for doing them at a simple way. And then the other major reason why we are really emphasizing on this aspect of doing it on the classical way and learning it out is that. So, that you can draw correlates between what happens on the deep learning side of it and what happens exactly on the classical way of it and also there are certain problems where you might not need to necessarily use a deep learning solution, although it is a buzzword it is its really interesting, but then it is a computationally heavy problem; and sometimes even simple tricks like we are doing over here might solve out all the major issues over there. So, I would try to keep down the course in a mix and match having both of these aspects covered down together and nonetheless.

(Refer Slide Time: 26:21)



The last part over here is to give you a take home message. So, if you would like to learn more about these local textures and classical ways of processing out images, then do definitely feel free to go through this book on image processing dealing with textures by Maria petro and Sevilla.

So, this is one of the classical books to do with it. So, with that we come to an end of these classical techniques for visual computing prospects. And eventually in the next lab we would be touching upon them, and then subsequent to that we would enter into classification with neural networks which is just a starting point of neural networks, and subsequently a lab to get down with this basic introductory a week.

Thank you.