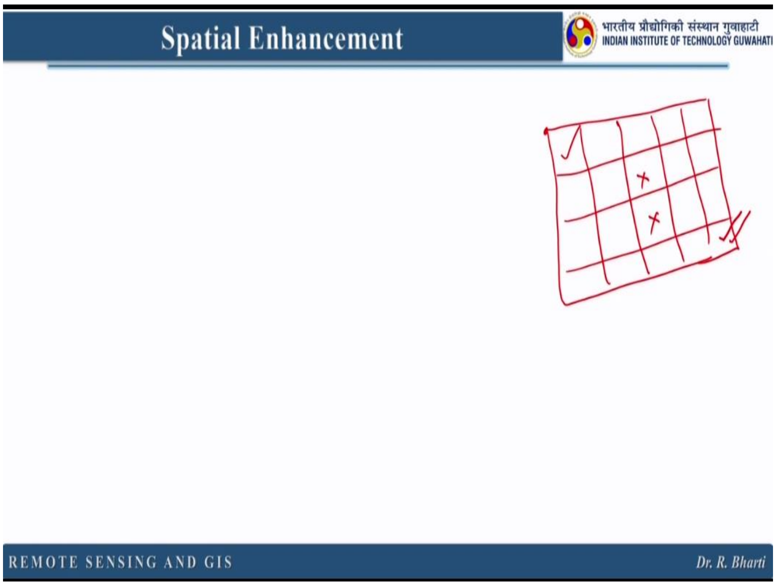
Remote Sensing and GIS
Rishikesh Bharti
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 08
Digital Image Processing (DIP) - II

This is the second lecture on digital image processing. So, I hope you have understood the concept which I have explained in the previous lecture. So, in this lecture let us start with the spatial enhancement.

(Refer Slide Time: 00:42)



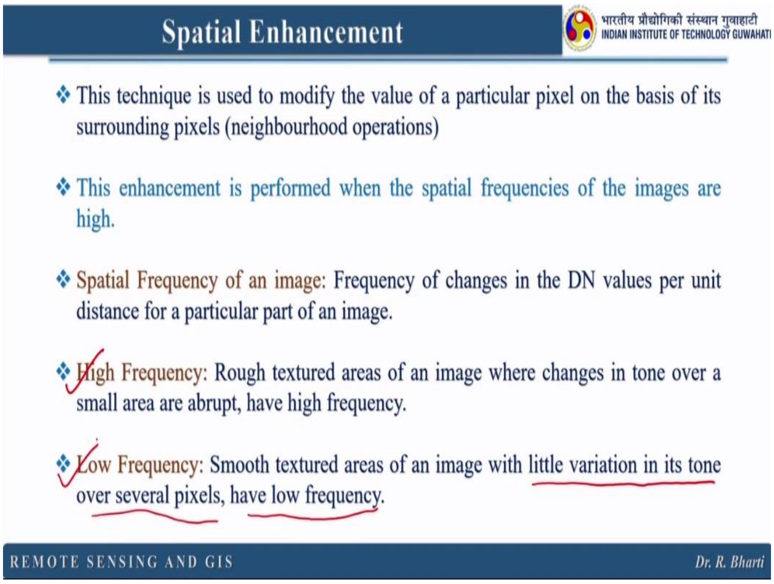
So, what do you mean by spatial enhancement? First of all, you have to understand like this is one image right and here there are many pixels right and these pixels are nothing but the DN numbers and whatever we do in image processing, we always do with the digital numbers and these figures which are visible to us through some screen or monitor that is only for the visualization purpose.

But actually, we are bothered about the digital numbers in terms of digital number, reflectance, emittance or scattered values. So, here what we mean by spatial enhancement? So, first of all, let us consider this is the first pixel of this image and this is the last pixel. So, how to define this first and last? So, first of all, we need to set our origin somewhere. So, suppose this is my origin.

So, this will be (1,1) right, so likewise we will move and we will find out what is the Cartesian coordinate for this particular last pixel of this image. So, here one thing is to remember that every pixel or every digital number is having 3 values right. One pixel is having 3 values, two are like latitude, longitude and another one is digital number in terms of like radiance, reflectance or emittance or back scattered energy.

But it is having 3 values, so x and y are basically the location values for this particular pixel. So, in this case, when we are talking about spatial enhancement, we always take care of their location. So, if you are not taking care of the location that means it will not qualify for this spatial term right.

(Refer Slide Time: 02:48)



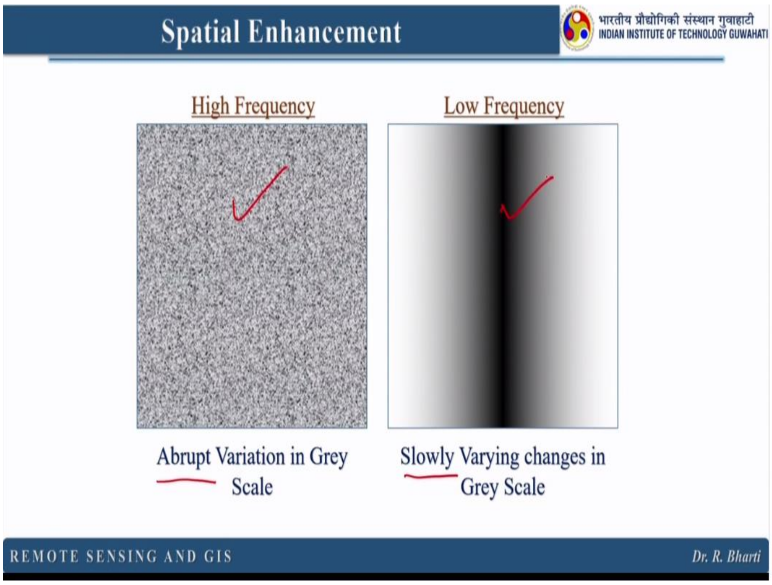
So, this technique is used to modify the values of a particular pixel on the basis of its surrounding pixels. This is very important because when we are going to start or when we are going to target a pixel for any kind of enhancement what we do, we have to do it with respect to their surrounding pixels, so in other word neighborhood operation right. So, this enhancement is performed when the spatial frequencies of the images are high.

So, here let us see what we mean by spatial frequency of an image. So, frequency of change in the DN values per unit distance for a particular part of an image. So, if this is your image right, so we are talking about only a particular portion or particular area of that image not for the entire image right. So, this is to remember. So, high frequency, when we say high frequency image right.

So, here rough textured area of an image where changes in tone over a small area are abrupt and have high frequency. That means if this is my image and it is having this kind of like behaviour right. So, what is happening here, here either we will have this let us assume this Boolean image 0 1 0 1 0 1 right. So, here the frequencies are very high right and in case of low frequency image what happens, smooth texture areas of an image with little variation in its tone over several pixels have low frequency.

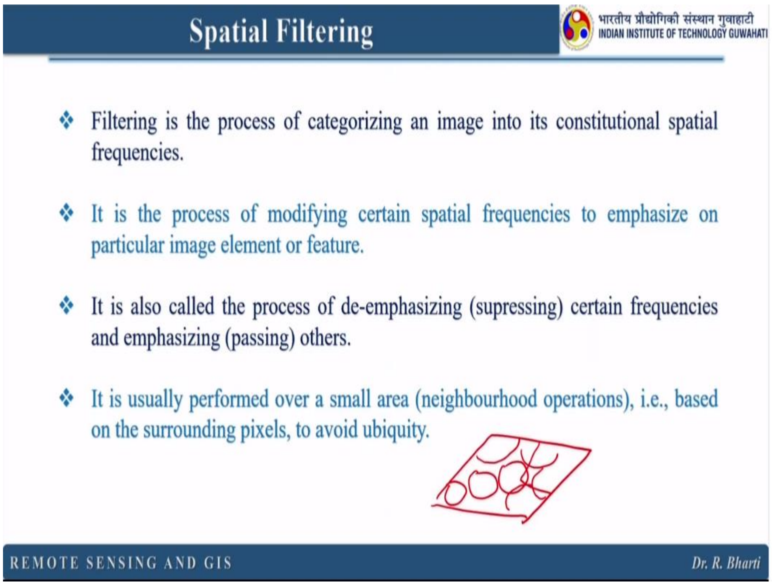
This is actually changing or it is classifying the image into 2 groups. First one is high frequency, another one is low frequency.

(Refer Slide Time: 05:09)



Let us see some examples. So, here this is one example where the tones are changing abruptly and in the second case it is changing slowly right. So, here this is abruptly and this is slowly. So, which one will be the high frequency image, this one right and this will be the low frequency image. Is it clear?

(Refer Slide Time: 05:33)

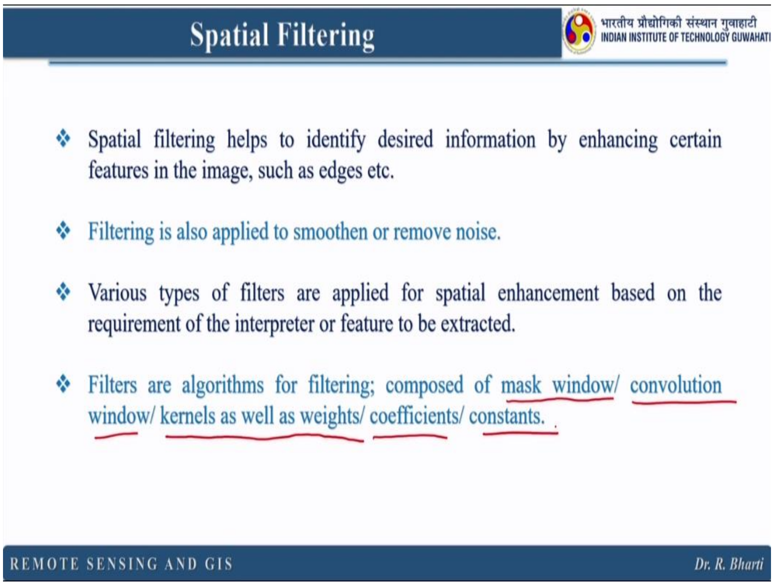


Now, what we mean by spatial filtering? So, filtering is the process of categorizing an image into its constitutional spatial frequencies like it is a process of modifying certain spatial frequencies to emphasize on particular image element or feature. So, here basically what we are doing, we are identifying a particular type of spatial frequencies which we want to highlight or either suppress or enhance by using some filtering mechanism right.

So, it is also called the process of de-emphasizing certain frequencies and emphasizing others right. Suppose if this is the area and you want to highlight this particular portion of this image, then what will happen, this will get enhance but this will get suppressed. So, it is working in both the direction right. It is usually performed over a small area that is based on the surrounding pixel to avoid ubiquity right.

Why? Because in a big image right, there are many portions and their spatial frequency or their frequency will be different. So, if you use a single logic to enhance this whole image, it may not work right but if you select a certain pockets or certain small area and if you apply a logic, filtering a logic then this image, output image will be very very good compared to its original raw data.

(Refer Slide Time: 07:28)



Spatial filtering helps to identify desired information by enhancing certain feature in the image such as edges. So, let us take one example like this is Mumbai right, this is the whole Mumbai right and here you want to highlight or you want to map all the major roads right. So, roads will be running parallel to each other or sometime they will be crossing to each other.

Then, there will be some NH, then there will be some state highways, then there will be some kaccha roads right. So, all these things you want to map. So, how you will do that because if you just consider all other areas right and with respect to that how we are identifying this particular road because in the boundary of this particular road, the contrast or the brightness is changing abruptly right but in other areas it is changing slowly.

So, suppose if I draw just figure, normal figures and why we are getting this particular perception that this is some kind of house or something why? Because of these lines, so these lines where intensity is changing abruptly. So, filtering is one of the mechanisms or one of the logic which can be used to identify edges. So, this is one example. So, if you want to highlight or you want to map the roads from a satellite data, you can simply use a filtering mechanism or filtering algorithm and then you can easily map the roads.

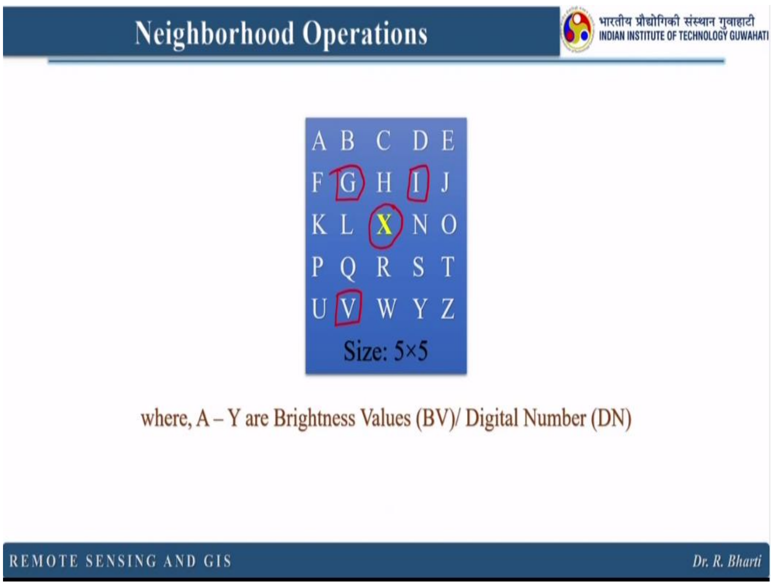
Likewise, there are many other examples. Filtering is also applied to smoothen or remove noise. You may remember from my previous lecture that sometimes we get pixels with bad values right and in case if you have missed to correct those pixel values, bad values then or if it is only one or two pixels which was not visible with your eyes and you could not find that, then if you apply this filter so definitely what will happen?

This will solve that problem but remember all these algorithm cannot give you the real value. These are the estimated values, so these values cannot be used for any quantitative analysis or in your quantitative research right. So, these are for qualitative analysis. Now, here it is also used for smoothen and remove the noise. Noise, I have given you this example and smoothen, suppose if you want to map the forest area of whole India.

Let us take this example and you have a very high resolution image, let us say 1 meter resolution image, which is not required to map the forest area right. So, then what you can do is you can smoothen the image, so your data will be of less file size. So, such things you have to take care when you are working with the data for a particular objective. So, next point is various types of filters are applied for spatial enhancement based on the requirement of the interpreter or feature to be extracted.

So, this is always you have to remember, this is based on the requirement and based on the interpreter or based on the application right. So, what exactly you want to identify. Filters are algorithm for filtering which is composed of mask window, convolution windows, kernel as well as weight, coefficient or constant right. So, how we will apply this, so that we will see in subsequent slides.

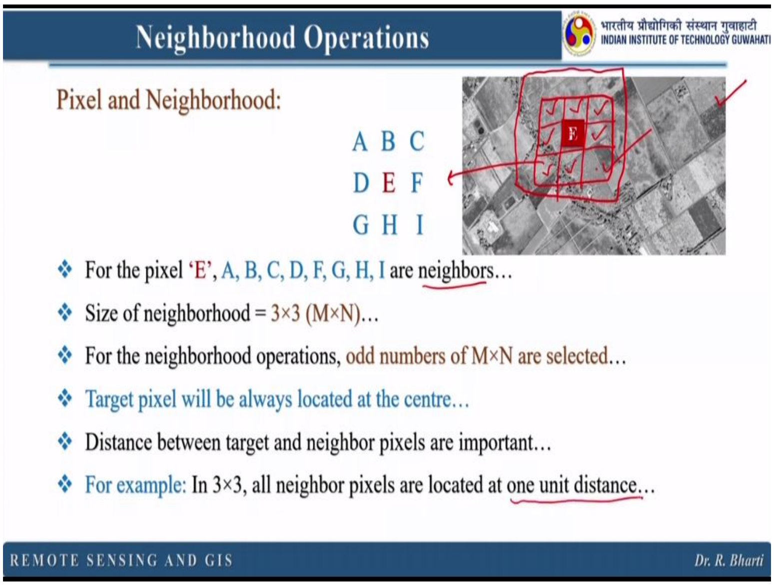
(Refer Slide Time: 12:02)



So, let us start this neighborhood operation. Basically, what we mean by this neighborhood in terms of image, so let us take this example where we have an image and these A B C D E F G H I up to Z are the digital number or the brightness value of this image. So, these are the associated values and here basically we are looking at the matrix right. So, here the size of this image is 5 by 5.

Now, we have to find out or we have to enhance the value of this X pixel, which is located in the centre. If it is located here what will happen, if it is located here what will happen, if it is located here what will happen, that we need to understand right. So, this is the detail of this.

(Refer Slide Time: 13:01)



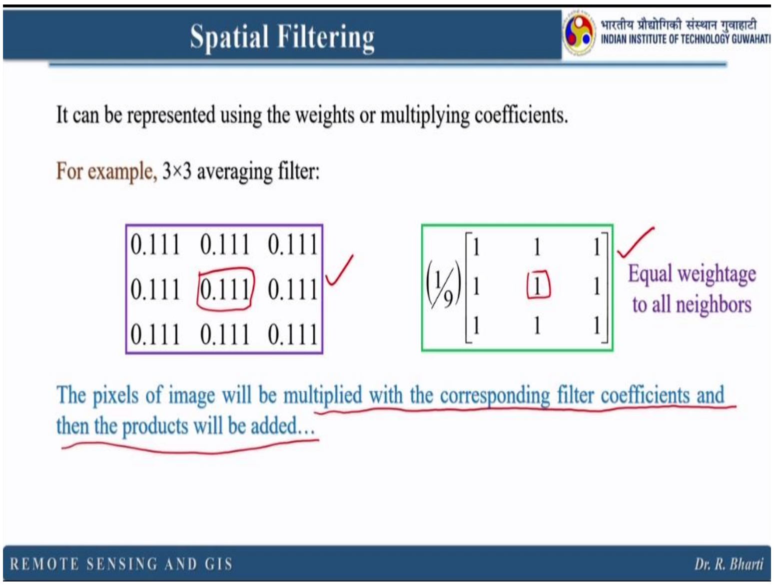
Now, pixel and neighborhood, so let us take this example. When you are having an image right, this is an image and you want to highlight or you want to run the neighborhood operation for this pixel E right and let us assume we have taken only this particular window right. Window means this is the extracted value of this particular area. So, here you will have this kind of right and this is represented here and in centre the pixel is E.

Now, we have to run this neighborhood operation on the pixel E, let us understand how we can do this. So, for the pixel E, A B C D F G H I are neighbors right and size of the neighborhood 3 by 3 right and for the neighborhood operation odd numbers of M into N are selected. So, if you want to increase this neighborhood size, so either it will be 3 by 3, 5 by 5 or 7 by 7. So, it has to be odd number.

Why? Because your centre has to be located at the centre, so that is the reason and the target pixel will always located at the centre, distance between target and neighbors pixels are important. So, in case of 3 by 3 matrix right, so this neighborhood size when we are taking 3 by 3 what is happening, the distance from E to this pixel, E to this pixel, E to this pixel, E to this pixel, everywhere it is at same distance right.

So, this is very important. When it is 3 by 3, all the neighboring pixels are at equal distance but when you are taking 5 by 5, 7 by 7 how it will grow? It will grow like this right for the same E pixel. So, this is in case of 5 by 5 and if you further grow, it will become 7 by 7. So, that is how we are going to apply the pixel and neighborhood concept in filter right. For example, in 3 by 3 all neighboring pixels are located at unit distance right. This already I have explained to you.

(Refer Slide Time: 15:47)



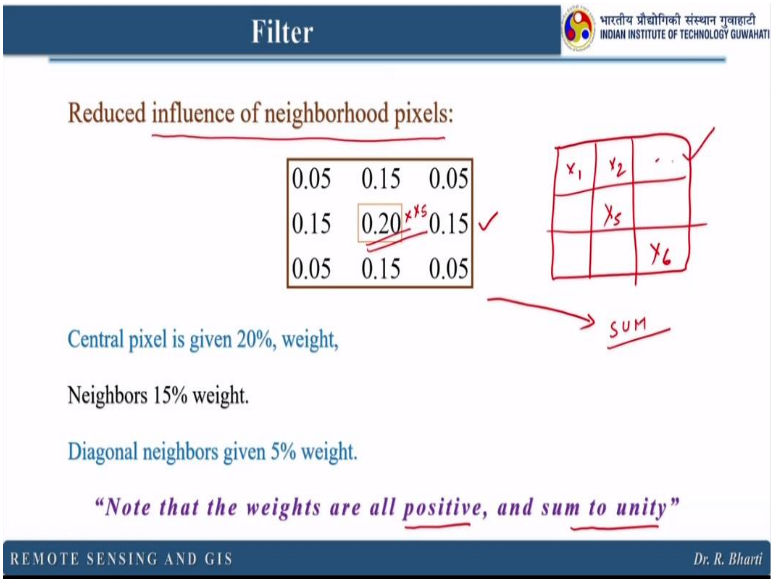
A spatial filtering, it can be represented using the weights or multiplying coefficient. So, here what happens, we have to apply some logic, how we will apply that we need to understand right. So, for example for 3 by 3 averaging filter right, we can use this kernel right or this is also known as weights or multiplying coefficient. So, just imagine you have an image right of 3 by 3, 3 by 3 pixel and it is having some value like 20 30 40 30 60 20 15 18 9 right and you want to calculate or you want to apply this averaging filter.

Then, either you have to apply this or this to this particular image. So, what we will do? We will exactly overlap this image pixel with this weights or multiplying coefficients right and then we will calculate the average. Then, what will happen? Here all the pixels, neighboring pixels including the centre one, they have got the similar weightage. So, this will not change any value of the given pixel, only the averaging will be performed.

But this is how we are going to perform right and here equal weightage to all neighbors have given right and the last point is the pixel of image will be multiplied with the corresponding filter coefficients and then the product will be added. There is the simple logic for averaging but here only thing we have added is the averaging filter either this or this and where all the neighboring pixels' values are not getting changed and we are simply doing the averaging for this central pixels.

So, in this case, there is a central pixel this is the, so these are weights or multiplying coefficients and your image DN values will be different.

(Refer Slide Time: 18:11)



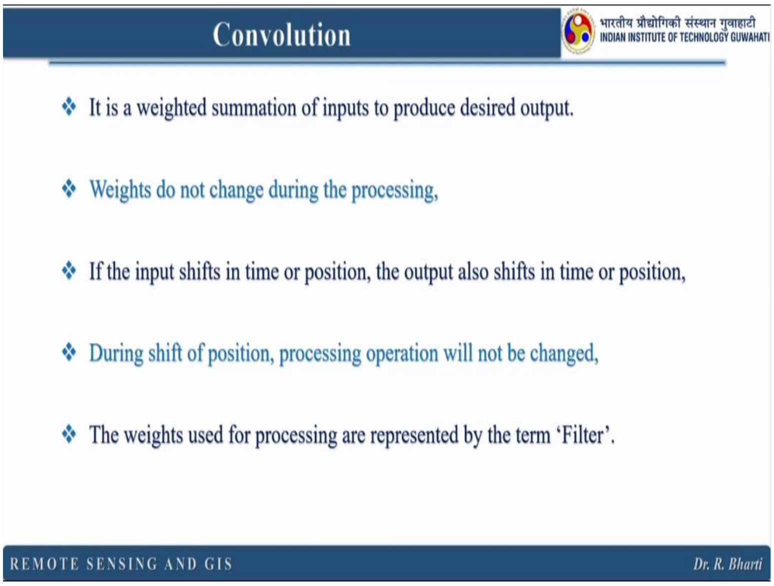
So, in the previous case, what we have seen, we have seen that we have given equal weightage or multiplying coefficient to all the neighboring pixel are same right but here what happens when I want to change the weightage, so reduced influence of neighborhood pixels. This is very important because always we do not go for the averaging. So, here let us see, this is your 0.05 0.15, so what is here, we have weightage for all the 3 by 3 pixel.

And here if you see the central pixel have got the maximum weightage. Why? Because we want to calculate or we want to perform this filter on the central pixel. So, the value whatever we will get that will be used to replace this particular value 0.20 right. So, here this is coefficient. Now, here you have image, let us say 3 by 3, so X_1 X_2 like that X_6. So, here this will be 3 4 5 X_5. So, this will be multiplied with X_5.

And then you will apply this and you will take the sum. Why sum? Because if you calculate all these coefficients or the multiplicative factors, this is actually equal to 1, so you do not have to do again the averaging. This is the weightage assigned to all these pixels, image pixels. So, here the central pixel is given 20% weightage right 0.2, then neighbors 15% weightage and diagonal neighbors are given 5% weightage only right.

And note that the weights are all positive and sum to unity. So, it cannot be in negative and it cannot be more than 1. So, that is the basic concept here that whatever design you want to have for your kernel or the weightage that you can follow but it has to be positive weightage and the total sum of the weightage it should be equal to 1 that is the criteria for these filters.

(Refer Slide Time: 20:54)

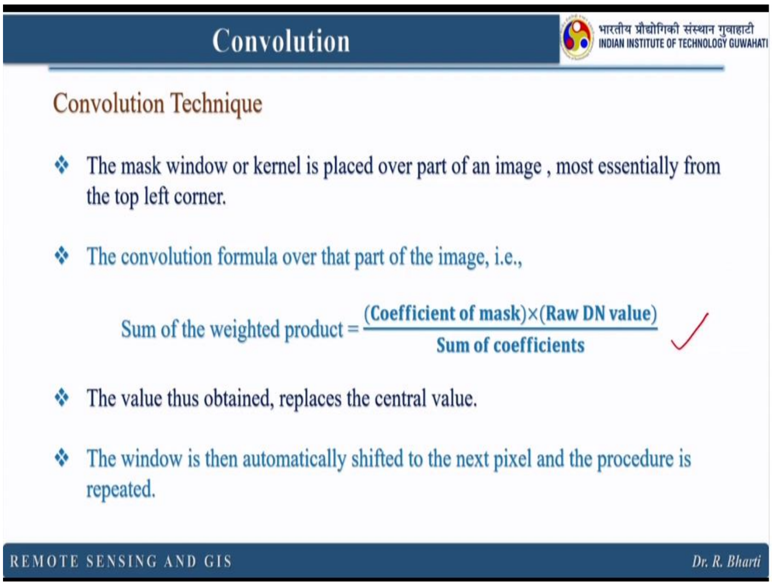


Now, let us see convolution. So, it is a weighted summation of input to produce desired output right. Weights do not change during the processing, so how do we perform this one that we will see in the next slide. If the input shifts in time or position, the output also shifts in time or position. So, here what it says you have an image and let us say this is 3 by 3 right. So, you have weightage with you that you are overlapping here.

And you are calculating the value for central pixel, then if you are changing the position of your image, your weightage will also move to that direction. Then, you will calculate the next pixel right. So, likewise you have to cover the whole image. During shift of position, processing operation will not be changed. So, here whatever logic you are using let us assume that averaging, then you should not change with any other logic for this particular image during this processing.

So, your logic will remain constant and you will move one by one to all the pixels and you will calculate the new values and the older values will be replaced with the new calculated values. The weights used for processing are represented by the term filter. So, here now we are calling those weightage as filter as together. So, it does not matter whether it is 3 by 3, 5 by 5 or 7 by 7, those are termed as filter right.

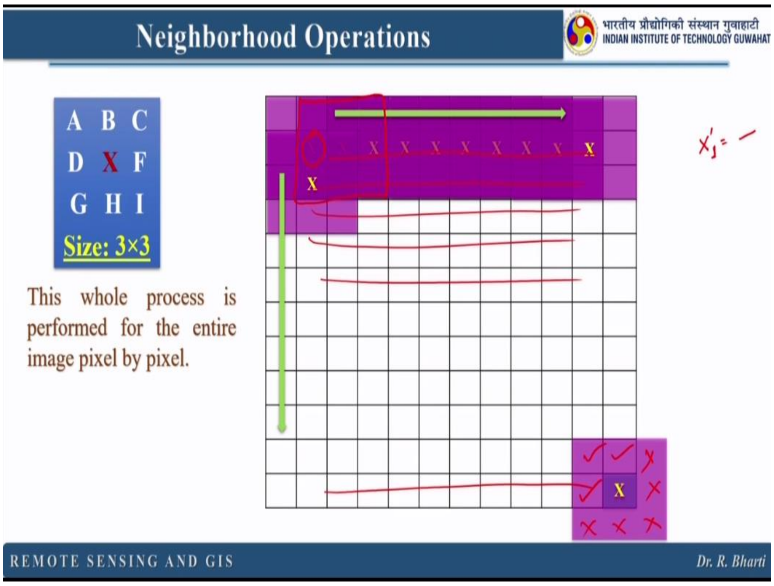
(Refer Slide Time: 22:49)



And then convolution techniques; so here we will see how we perform this convolution, the mask window or kernel is placed over part of an image, most essentially from the top left corner right. So, if this is your image, you will start from here and let us assume this is 3 by 3 right. So, this is what first point says. The second point is the convolution formula over the part of the image right we are putting our convolution formula on top of the image.

This is just to understand how we perform this and then we use this particular logic. The values thus obtained replaces the central value as I told you that one by one we will calculate for all the pixels and the new calculated value will be used to replace the older values. The window is then automatically shifted to the next pixel and the procedure is repeated. So, this is how we are going to cover the whole images.

(Refer Slide Time: 24:03)



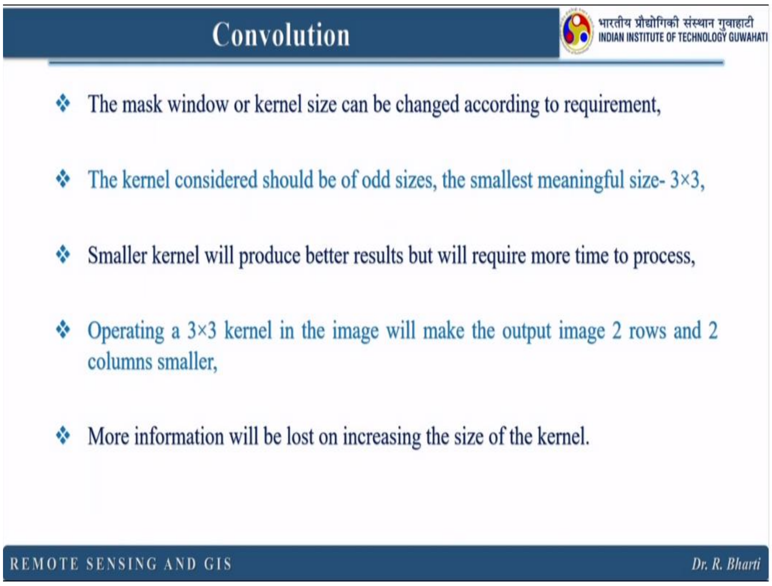
So, let us see this one example where this is the size of your kernel right. So, this is the size of your kernel where X is your central pixel and A B C D F G H I these all are basically weightage right. So, now what we will do, first let us decide what size of the kernel you want to apply. So, here in this case, we are using 3 by 3, so 3 by 3 means 1 2 3 4 5 6 7 8 9 so basically this is my central pixel.

So, this central pixel value or the new value will be calculated now. So, how we will do that, so let us assume that you have overlap. So, this is the central pixel. Now, using this let us assume that this is average filter right and you have calculated new X. So, X dash or X1 right and this is some value. Now, you will move to next pixel, then again this is your new neighborhood right.

And again further you will move, again you will calculate the new value, likewise you will calculate for all other pixels. Is it clear? The whole process is performed for the entire image pixel by pixel. So, here what I mean to say, we have covered this, then we will cover this, then this, this, this like that we will cover the entire pixel and the new pixel value will be used to replace the older pixel value.

But just assume what is happening with the corner pixel. So, here you do not have neighborhood right. So, all these values are present but these values are not present. So, what will happen to these pixels right all the corner pixels or the boundary pixels that we will see?

(Refer Slide Time: 26:26)

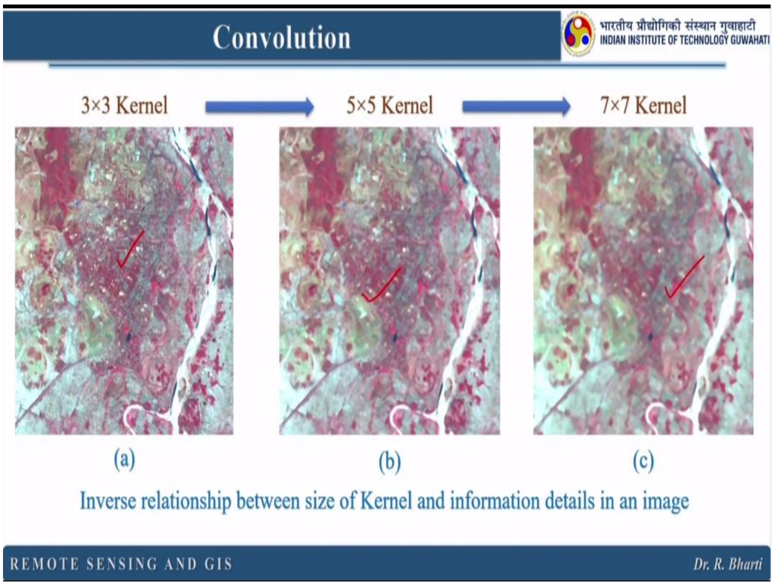


The mask window or kernel size can be changed according to requirement. So, here what is your requirement whether you want more smoothing or less smoothing that tells you whether you are supposed to use 3 by 3, 5 by 5 or 7 by 7. So, the kernel considered should be of odd size, the smallest meaningful size is 3 by 3 and then further you can increase it to 5 by 5, then 7 by 7 like this you can go on.

But depending upon your application you are supposed to use or identify or decide the size of the kernel. The smallest kernel will produce better result but will require more time to process because we have seen in the last slide that if it is 3 by 3 moving one by one then it will take more time because the size is a smaller and then you will require more time. Then, operating a 3 by 3 kernel in the image will make the output image 2 rows and 2 columns smaller.

How? Because you are going to miss the data for the boundary pixels, so that was the problem I told you in the previous slide. We will see what are the solutions. Then, more information will be lost on increasing the size of the kernel because if you increase by 7 by 7 then what will happen again you do not have pixels for the boundaries, then what is going to happen?

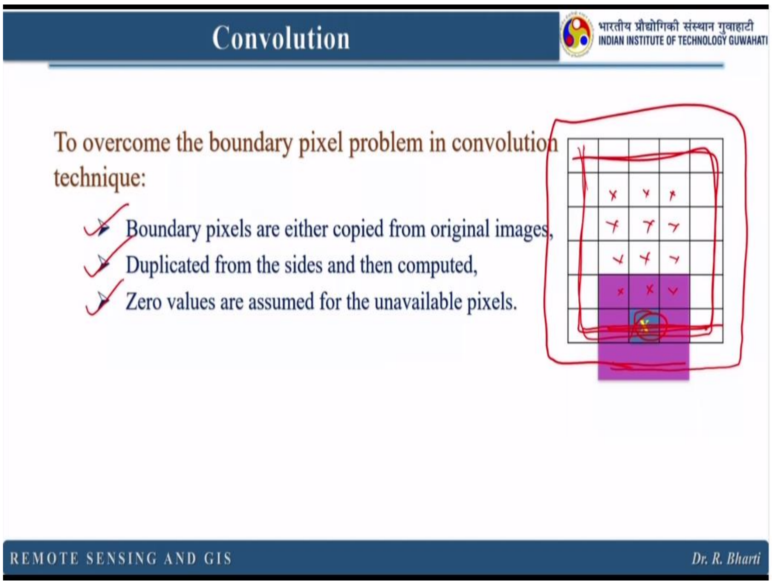
(Refer Slide Time: 28:06)



So, this is one example when you are using 3 by 3 kernel, 5 by 5 and 7 by 7 kernel. So, this is one example when you are using averaging. So, here in 3 by 3 you can see details but here it is little bit blurred, why because it has been averaged using 5 by 5. If it is 7 by 7 then again it is more blurred, why because you are using more pixels to calculate the average. So, the boundaries are the change in the intensity will not be preserved.

So, it is again the inverse relationship between size of kernel and information detail in an image. So, here you need to decide or you have to do the tradeoff which one you will prefer for your application.

(Refer Slide Time: 29:00)

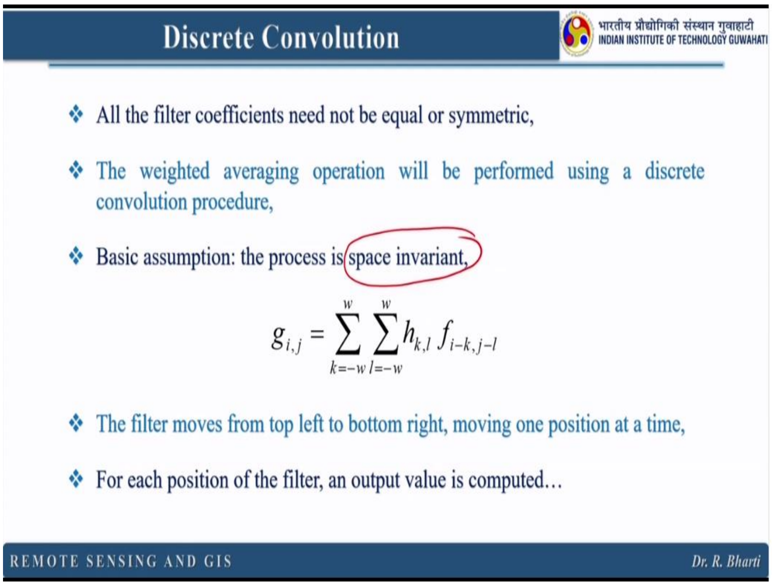


So, let us see for this particular pixel the boundary pixel what is going to happen and how we are going to solve this problem. So, to overcome the boundary pixel problem in convolution technique either we have to use boundary pixels or either copied from original images, so we will not calculate anything, we will just copy these values in the output one because what we are doing, we are calculating these pixels value right.

So, here in this case only these pixel values will be calculated if you are using 3 by 3 and then all these boundary pixels value will be copied from the original image that is one solution right so that you will have at least the original data. In the next one, duplicates from the sides and then computed. So, duplicates of the sides means this pixel value will be copied here and then you can calculate this central pixel value, this is another possibility.

And third possibility is put zero values in the boundary right and then calculate these boundary pixels value. These are the 3 possibilities but depending upon your application again or the interpreter he or she has to identify which method works best for you right.

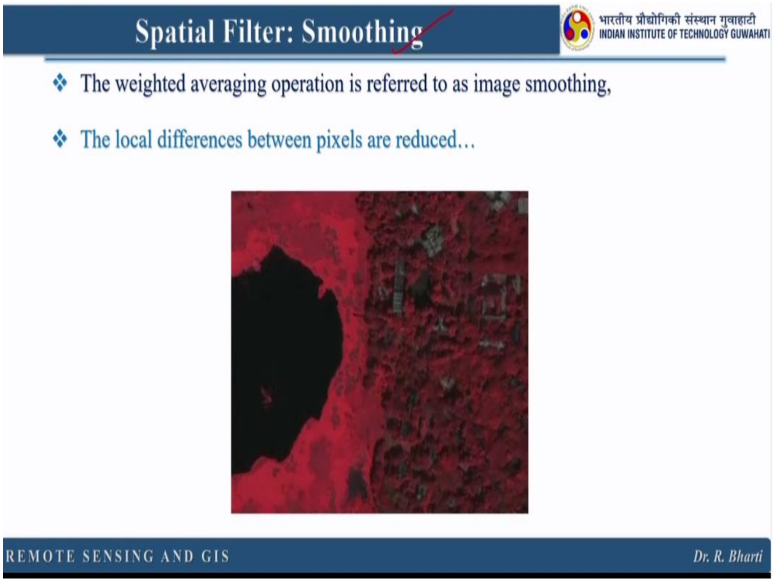
(Refer Slide Time: 30:45)



Now, discrete convolution; so all the filter coefficient need not to be equal or symmetric. So, in the previous case we have seen when we are using similar weightage for all the pixels but here it need not to be equal or symmetric so what we can do, we can change. The weighted averaging operation will be performed using a discrete convolution procedure. So, how we will do that?

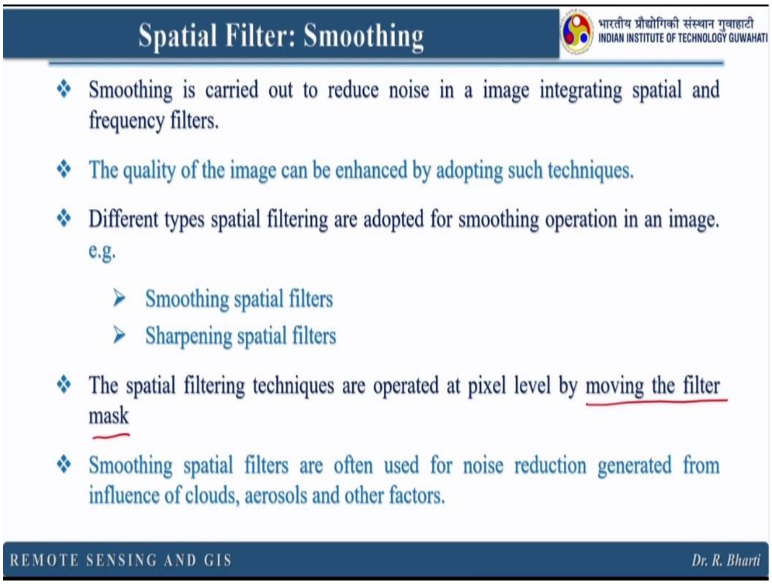
So, basic assumption is the process is space invariant. So, this is very important, space invariant. The filter moves from top left to bottom right, moving one position at a time. For each position of the filter and output value is computed. So, here we are assuming that this process is space invariant and it is going to apply on this particular image.

(Refer Slide Time: 31:47)



So, in case of smoothing what we do is the weighted averaging operation is referred to as image smoothing. This is the basic definition. Now the local differences between pixels are reduced when you are using this smoothing right. Here you can see some example like this is one raw image and if you apply a smoothing what will happen right. You can see, I hope this is visible to you right. So, here difference in the intensity will be reduced.

(Refer Slide Time: 32:31)



Smoothing is carried out to reduce noise in an image integrating spatial and frequency filters. So, now you can remove noises from this image from your target image and by integrating spatial and frequency filters. So, what are they, we will see slowly all the things. The quality of image can be enhanced by adopting such techniques because here our purpose is to enhance or to process our image and make it suitable for my application right.

So, why we are doing this, to enhance the quality of this image and by adopting such techniques which we are going to discuss. Different types spatial filterings are adopted for smoothing process in an image right. For example, smoothing spatial filter, sharpening spatial filters. The spatial filtering techniques are operated at pixel level by moving the filter mask that I hope this is clear to you that we have already covered this part by moving the filter mask.

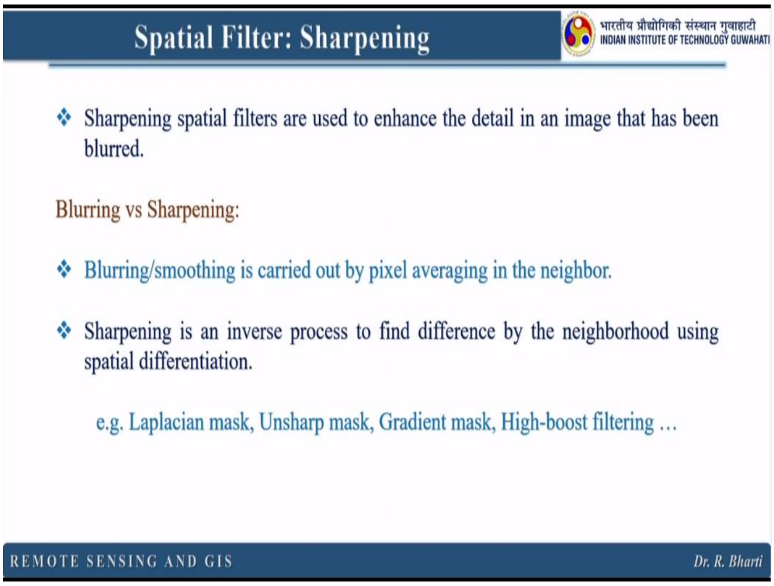
Smoothing spatial filter are often used for noise reduction generated from influence of clouds, aerosols and other factors. So, we have already seen what are the different types of or source of errors in remotely sensed image. So, somehow if there are some errors which are left and if you want to remove them so here this is very convenient way.

(Refer Slide Time: 34:23)

So, classification of spatial filter; the smoothing spatial filters are basically classified into two groups such as linear smoothing spatial filter, non-linear smoothing spatial filters. So, linear and non-linear behaviour I have already covered in previous lectures right. So, here the same logic applies and what do you expect that which one will be best, always in nature we have non-linear behaviour.

So, always we will have better performance of non-linear smoothing spatial filters compared to linear smoothing spatial filters. So, linear smoothing spatial filters are average of pixels contained neighborhood of filter mask right. For example, Mean, Weiner and Gaussian filters right. Similarly, non-linear spatial filter is based on ranking the pixel contained in the image area, so these are minimum, maximum and median filters. So, we will see the examples which one are best among these linear and non-linear filters right.

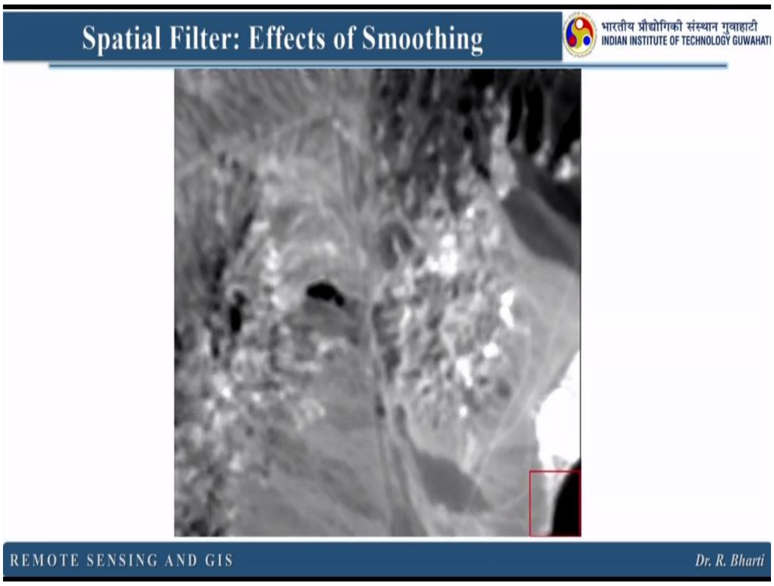
(Refer Slide Time: 35:44)



There are few different concept like sharpening, sharpening of the image. What do you mean by sharpening? Sharpening spatial filters are used to enhance the details in an image that has been blurred while acquisition right but you want to have very fine detail or enhanced detail of the edges. So, then you will say that it is better or image has been sharpened right. So, blurring versus sharpening, both are inversely proportion to each other.

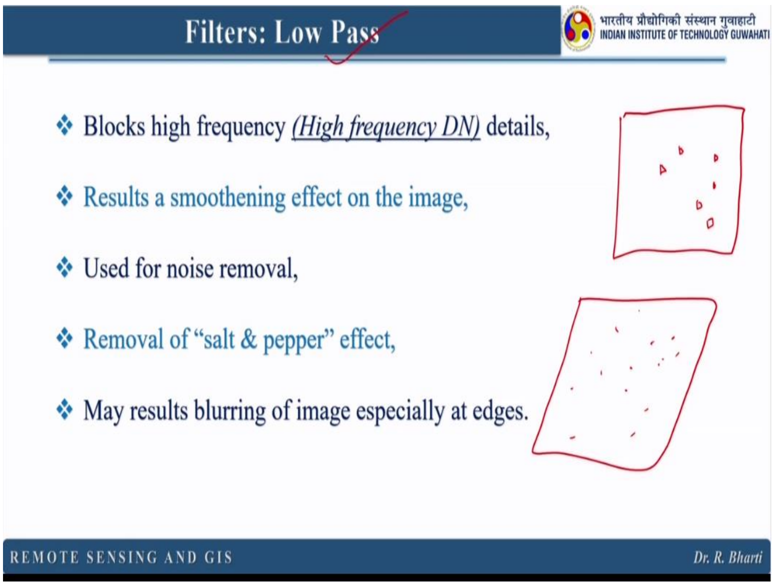
Blurring or smoothing is carried out by pixel averaging in the neighbor right that you have already seen few examples. Sharpening is an inverse process to find difference by the neighborhood using spatial differentiation right. So, for example Laplacian mask, unsharp mask, gradient mask, high-boost filtering. We will see all of them one by one, not may be in this lecture but in next lecture we will cover everything.

(Refer Slide Time: 36:54)



Then, what is the effect of smoothing on image that you need to understand very carefully. This is another example. Here when we are using 3 by 3 this is the example when we are using 5 by 5 and this is the example when we are using 7 by 7. So, here what is happening basically we are losing the edges right and pixel intensity variation are getting reduced. So, when you are having very less difference in the pixel values, what will happen, you will have smoothing effect. This is what evident from this particular image.

(Refer Slide Time: 37:39)



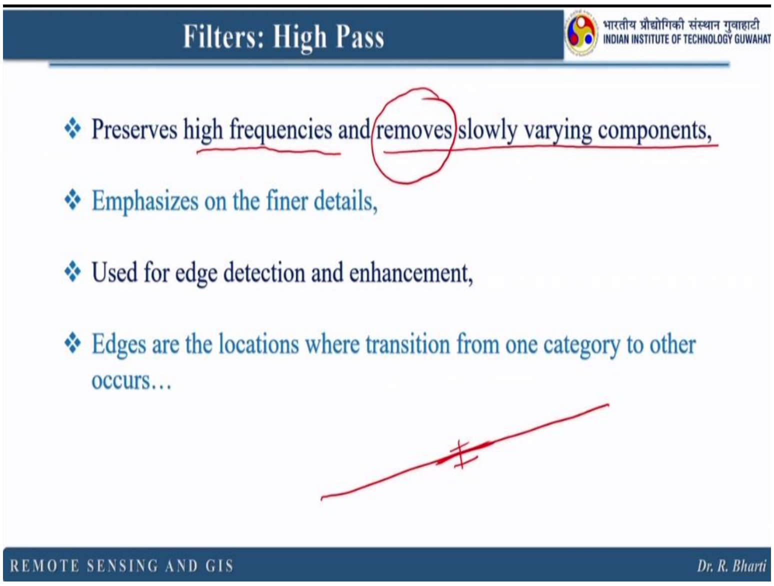
So, let us see in filter we have low pass filter and high pass filter. So, what do you mean by low pass filter. So, low pass filter blocks all the high frequency DN values right, so high frequency details means the image you have right, so there will be few DN values which will occur very frequently right. So, those DN numbers are basically high frequency DN values. So, those will be blocked here.

Then, it results a smoothing effect on the image. So, here first you need to find out high frequency DN values and low frequency DN values and then if you apply low pass filter, then all the high frequency values will be suppressed and low frequency values will be passed. So, your image will have different output than your input image that we will see some of the example and it is used to remove noise.

Why, because in between if you have one or two pixels, which have bad data or bad value, then those pixels will be corrected because it is having the smoothing effect and it is always preferred in salt and pepper noise. So, some of the time in your image, you may find there are few pixels with bad values right and it is something like you are putting salt and pepper in a white paper right.

So, those effects will be removed when you are running the low pass filter and may result blurring of the image especially at edges. So, this is the limitation of this low pass filter but again we can overcome from this problem by using the original image DN values in the boundary pixels right and there are different methods which we have already covered. Next is high pass filter.

(Refer Slide Time: 40:01)

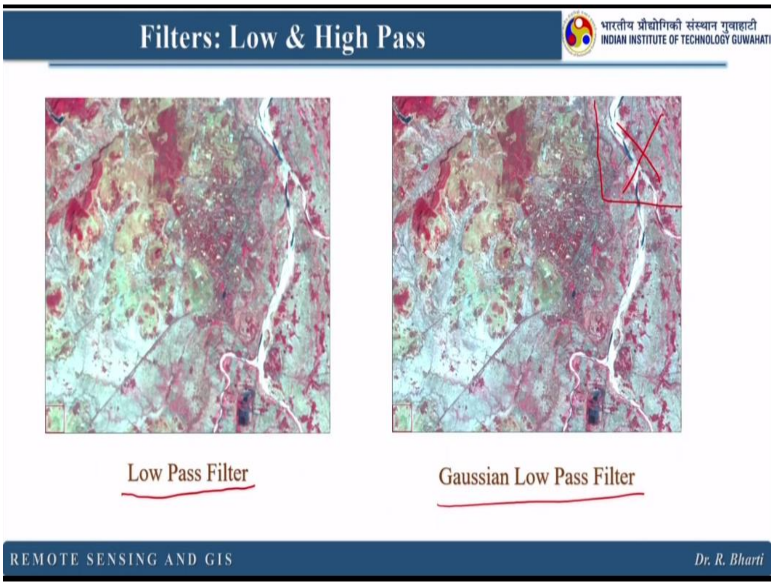


In high pass filter, as I told you that it will allow high frequency DN values and it will suppress low frequency DN values. So, what will happen, it will be inverse of your low frequency or low pass filter. So, it preserves high frequencies and removes slowly varying components. So, it will remove or it will suppress right. The next is emphasizes on the final details because the high frequency DN values are the DN values which are found in the boundaries or edges right.

So, if you have preserved your edges, you will have more detail about the data that we will see that there are many other filters which are used for age detection and enhancement. Edges are the location where transition from one category to other occurs. So, basically I told you that when I am drawing a line, how we are pursuing this line because there is a sudden change in the intensity from this to this right.

So, along this line the intensity either it is increased or decreased and because of that only we are perceiving these edges right.

(Refer Slide Time: 41:24)

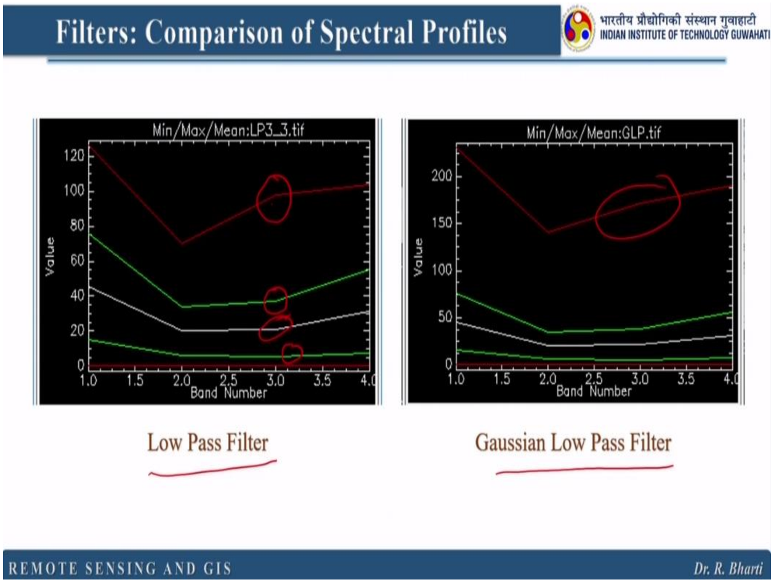


Let us see some example of low and high pass. So, this is the result from low pass filter and this is the result from high pass filter. So, as we discussed in the previous slide, high pass filter will preserve the edges. So, it is better when you are going to study or when you require the edges of your study area, you are supposed to use this high pass filter where all these edges are preserved and enhanced and all other information are suppressed.

But here what is happening, your image is getting blurred right because of the effect of smoothing. This is another example from low pass filter and this is Gaussian low pass filter. So, here in low pass filter basically we are allowing the low frequency DN values to pass and high frequency DN values are suppressed but in Gaussian low pass filter, the basic assumption is your low frequency DN values are equally distributed in your image.

Then, only it will give you good result but in case if all the low frequency DN values are concentrated in this particular area and then you are applying this Gaussian low pass filter, then the result may give you wrong information right.

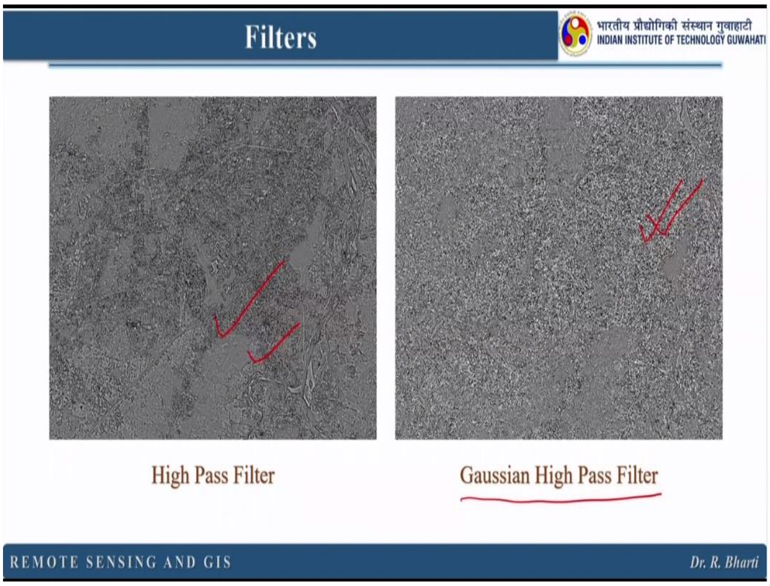
(Refer Slide Time: 42:58)



So, this comparison we can see when we are comparing the spectral profile of the same pixel with low pass filter and another is Gaussian low pass filter. So, here in low pass filter basically this particular P you can see in all the bands this is preserved right but here when we are applying Gaussian low pass filter, this is basically disappeared. So, why it happened because my image was not having Gaussian distribution of low frequency DN values.

So, you have to be very careful when you are selecting a Gaussian low pass filter for your image. Yes, now in which condition you can apply this Gaussian low pass filter, when your image size is very small. So, for big area, it may not work but in small area it may give you better result than this low pass filter.

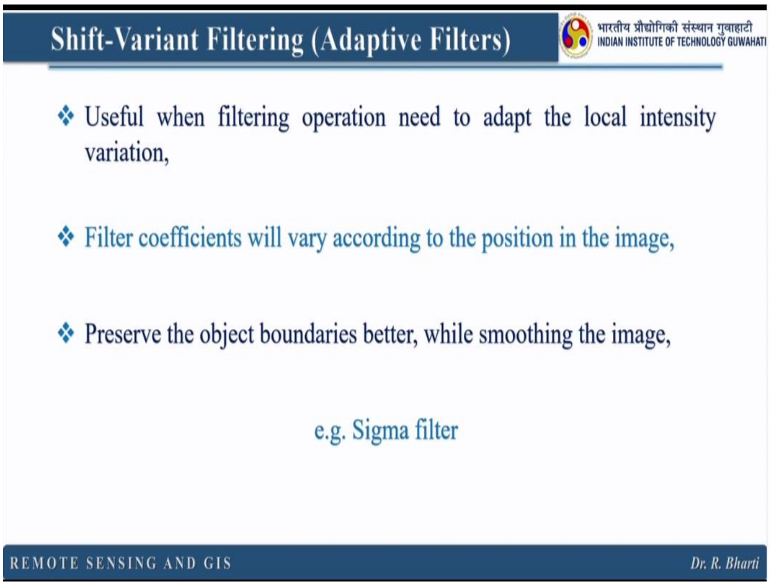
(Refer Slide Time: 44:08)



So, now we have example of high pass filter and Gaussian high pass filter. So, you have already seen this image. So, this is the result of high frequency DN values passed from this filter and this is when we assume that our high frequency DN values are distributed equally in whole image but when it is not, then because of that bad assumption, this is the result.

So, here you can see that this image has no information or very less information compared to high pass filter or compared to your original data. So, you have to be very very careful while selecting any of these filters in your application.

(Refer Slide Time: 44:56)

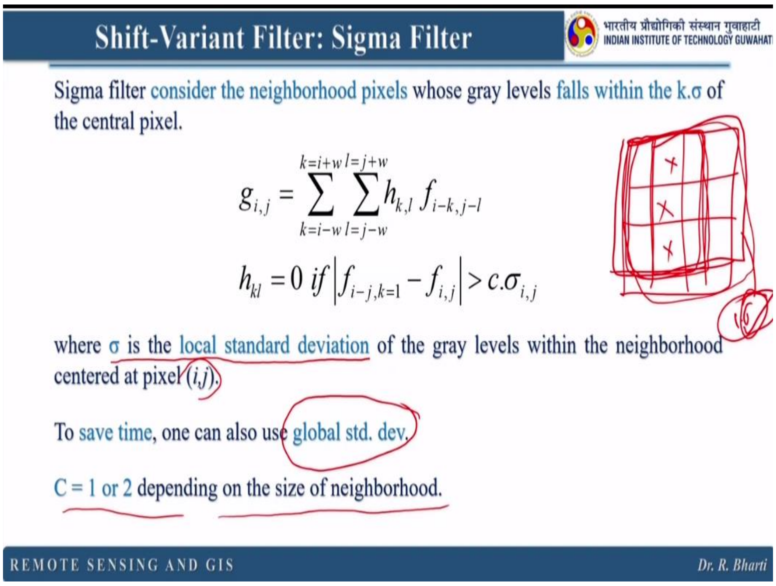


So, in the previous cases, you have seen that the local variation in the intensity were not preserved because of that you are getting more noise after applying those filters. So, there are few filters like shift-variant filter or we also called them adaptive filter. So, these are used when filtering operation need to adapt the local intensity variation right and filter coefficient will vary according to the position in the image.

So, what it says like this is the image right and this is my coefficient. So, here I am assuming 3 by 3 and this is the image right. So, what I will do, I will overlap this to this particular image and then I will calculate the value of this particular pixel. So, here in the previous cases, these weightage were constant but here in these shift variant filters what we do based on their location and the DN values, the weightage may change right.

So, how we will change them? So, based on different statistical measures. So, it preserves the object boundaries better while smoothing the image. So, for example sigma filter. So, here let us see what exactly we are doing when we are applying sigma filter to our image.

(Refer Slide Time: 46:38)



Sigma filters consider the neighborhood pixels whose gray level falls within the k sigma of the central pixel. So, here k may be 1, 2 or 3 and depending upon that which gray labels or which DN values are similar or coming within that range, those will be considered in averaging right. So, here this is the detail of this sigma, so basically it is the local standard deviation of the gray levels within the neighborhood centered at pixels like i, j right.

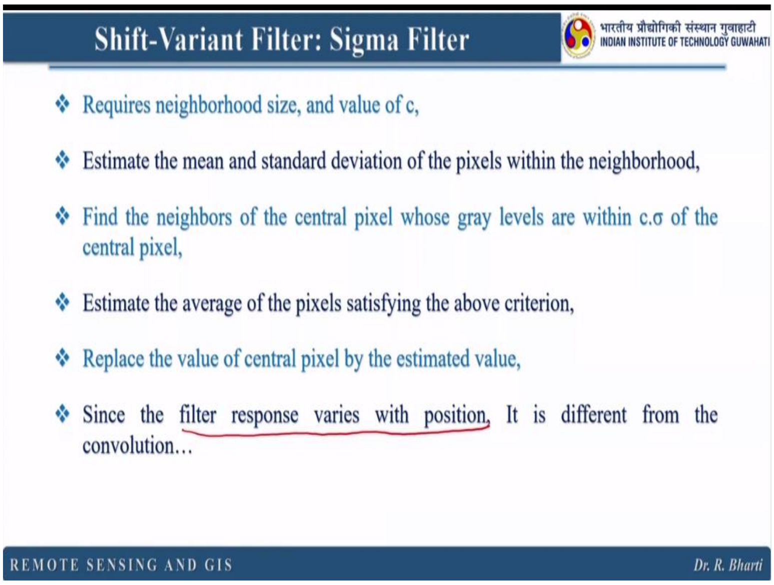
So, here you have one image and you have considered let us say 3 by 3 and this is the central pixel and then you have decided 1 sigma or 2 sigma and suppose only 2 pixels are qualified to be within this range, then only those 2 pixels will be considered in averaging right. So, to save time, one can use global standard deviation. So, here what is happening? Here based on this, we have calculated this standard deviation right.

So, 1 standard deviation, 2 sigma or 3 sigma that is up to us but how this sigma is calculated based on this 3 by 3 matrix. So, what happens next? Again, it will move to next 3 by 3, again it will calculate sigma and then but your 1 sigma, 2 sigma concept will remain same throughout this operation right but instead of calculating this 3 by 3 matrix standard deviation, one can use global standard deviation.

And which is calculated from this whole image right. So, this is to save the time because every time you have to put your kernel on the image then you extract those 3 by 3 pixel values, calculate the standard deviation, decide what is the k value and then those pixels which are qualified to be under that 1 sigma or 2 sigma, they will be used for averaging. Then, again you are shifting this to next position; then again you have to calculate the standard deviation.

So, here this is a time taking process, so instead of doing individually 3 by 3 for the whole image, we can use or one can use global standard deviation right and C=1 or 2 depending upon the size of the neighborhood. So, it is up to the interpreter or analyst what C value he or she is using.

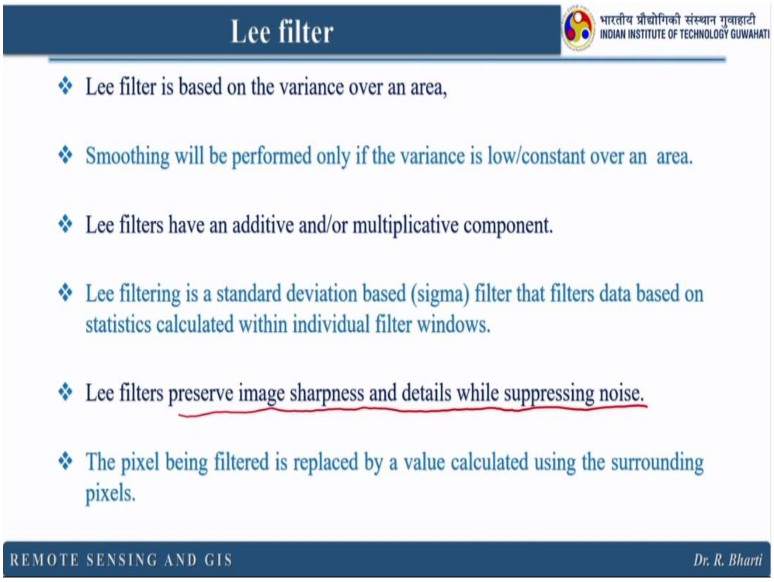
(Refer Slide Time: 49:40)



Further it requires neighborhood size and value of C. So, neighborhood size is either 3 by 3, 5 by 5 or 7 by 7 and C is 1 sigma, 2 sigma or 3 sigma and it estimates the mean and standard deviation of the pixels within the neighborhood, find the neighbors of the central pixels whose gray levels are within C sigma of the central pixel right. Estimate the average of pixel satisfying the above criteria and which is replaced or which calculated or estimated value is used to replace the values of the central pixels right.

And likewise, we will perform this for the whole image right. Since, the filter response varies with position; it is different from the convolution. Remember, this is not convolution why because filter response varies with position, this is very important.

(Refer Slide Time: 50:48)

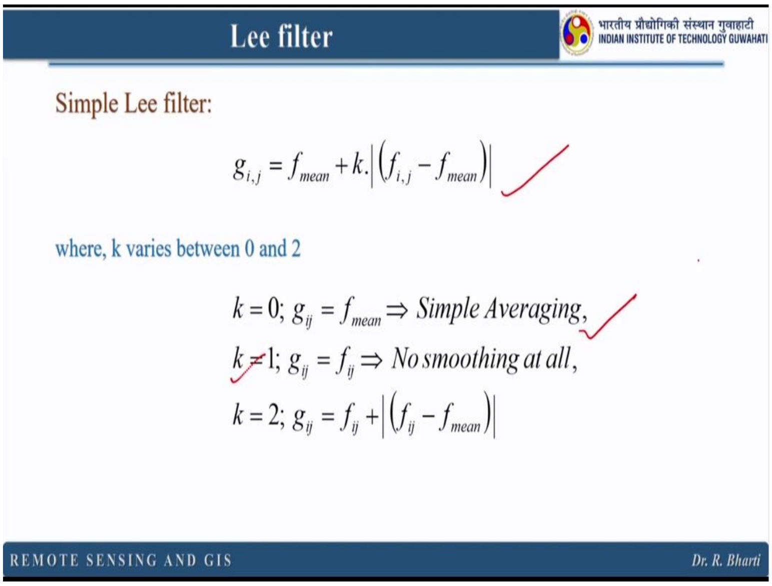


So, next filter type is Lee filter. So, here the Lee filter is based on the variance over an area. So, here we are using variance right. Smoothing will be performed only if the variance is low or constant over an area. Otherwise, this method will not be applied there right. Lee filter have an additive and/or multiplicative component, so that we will see. Lee filtering is a standard deviation based sigma filter that filters data based on the statistics calculated within individual filter window.

Again, we are doing this window by window. Window by window means in case you have considered the neighborhood size 3 by 3, then you are moving one by one to all the positions right and then you will calculate the desired central pixel value. So, here Lee filter preserve image sharpness and details while suppressing noise. So, here this is very good when you want to suppress the noise and you want to increase the sharpness of the image.

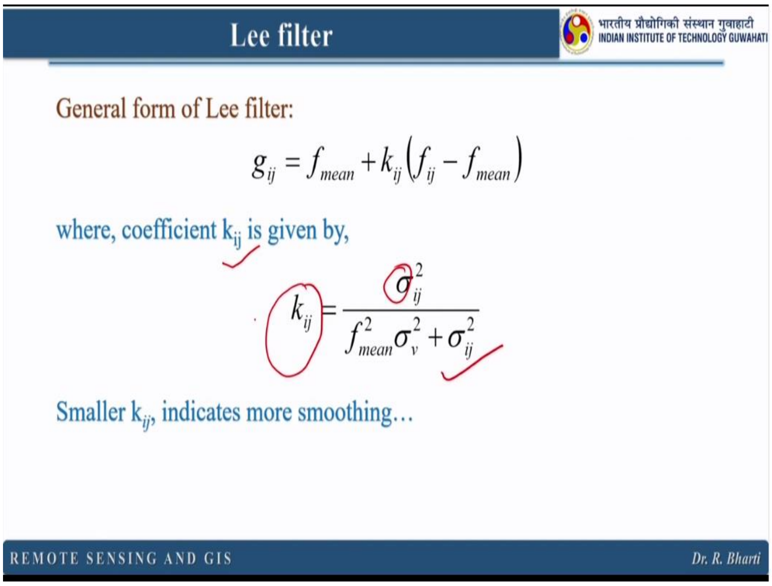
This is very good method. The pixel being filtered is replaced by a value calculated using the surrounding pixels. That is similar like similarly we were doing for all other methods. We calculate the target pixel, new value and then that will be used to replace the raw data value right. Raw data means image pixels without any calculation.

(Refer Slide Time: 52:34)



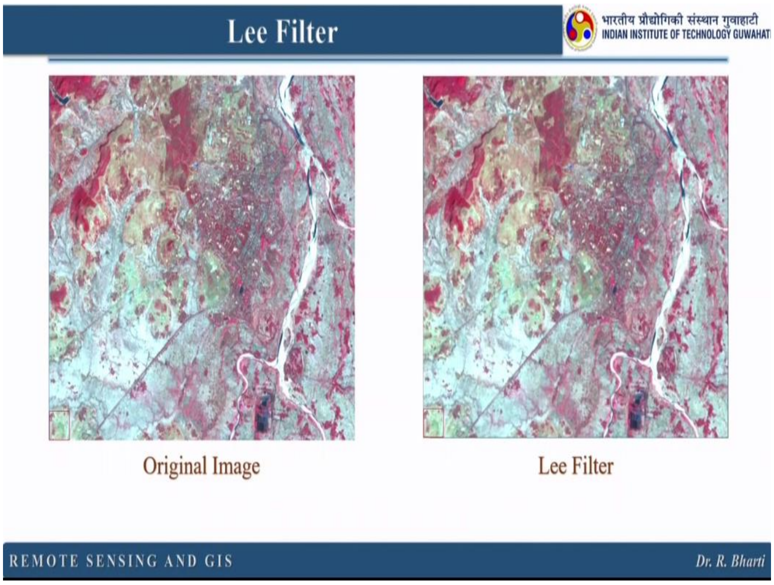
So, this is the general expression of simple Lee filter where k varies between 0 to 2 right and when you are using k=0 it is simple averaging and when you do not want to do any averaging, then you have to use k=1 right. So, this is the simple Lee filter.

(Refer Slide Time: 52:59)



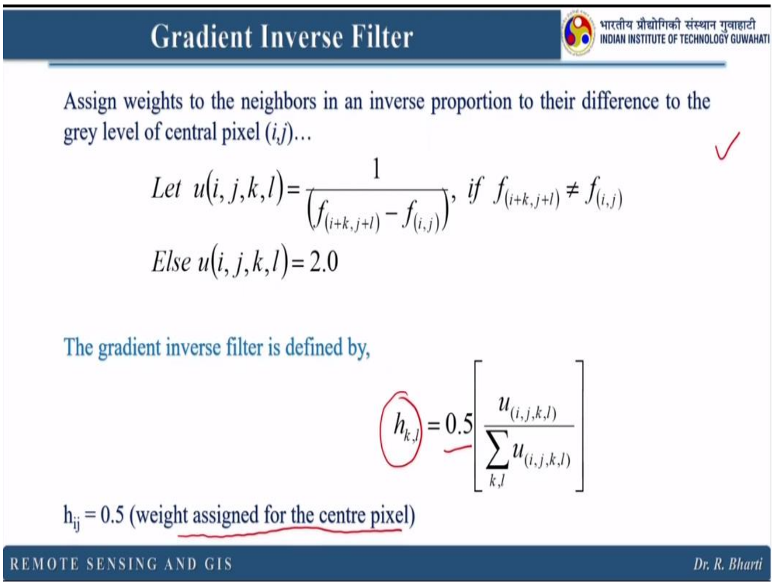
Now, general form of Lee filter is basically this one right where coefficient k_{ij} is given by this right where we are using standard deviation and smaller k_{ij} indicates more smoothing. So, smaller k_{ij} value indicates more smoothing. So, this is very important when you are going to run this particular method on your image, you have to make sure that this value is higher when you do not want to smoothen your image right.

(Refer Slide Time: 53:32)



So, this is the original image and this is the Lee filter image. So, you can see that in this case there were some features which were enhanced and noise were removed but it is up to you and your application, what exactly is your application based on that you have to select different methods from this particular basket right.

(Refer Slide Time: 53:59)



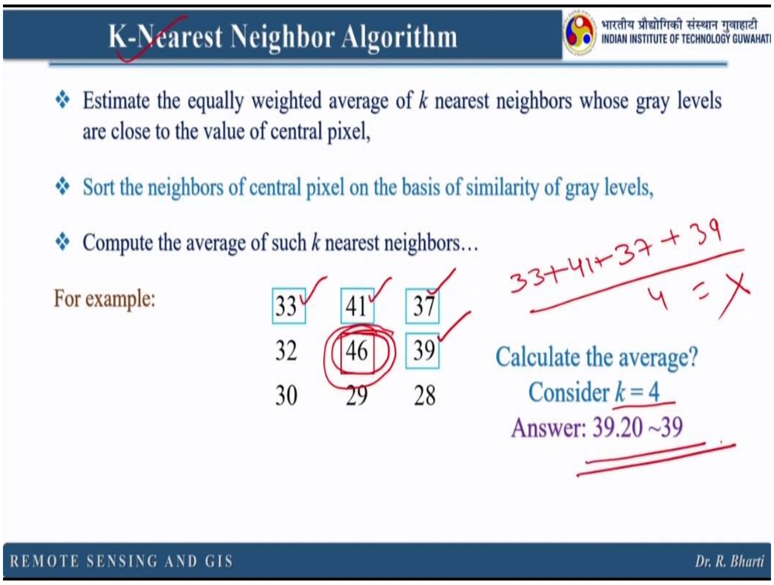
Gradient inverse filter; so it assigns weightage to the neighbors in an inverse proportional to their difference to the grey levels of the central pixel. So, suppose if you have a 3 by 3 image right and here the value is 10 and here 40 20 13 16 17 80 40 and 30 right. Suppose, this is one example where you want to apply this gradient inverse filter. So, here how do we assign the weightage?

So, it assigns weightage to the neighbors in an inverse proportional to their difference in the grey levels of the central pixel. So, here more weightage will be assigned to the values which are closer to this central pixel value right and less weightage is to all those values which are basically very different from the central pixel right. So, this is the general expression of this gradient inverse filter.

The gradient inverse filter is defined by this h k, l where this is 0.5 right and 0.5 is weight assigned for the central pixel and rest pixels are basically 0.5 right, so this is very important. In gradient inverse filter, you are giving more weightage to the values which are similar to your central pixel value and you are giving 50%, out of 100 you are giving 50% weightage to the central pixel.

So, here you want to preserve your original information right but at the same time you want to apply this gradient inverse filter to correct your image right. Next one is k nearest neighbor algorithm where what we do, we identify k nearest neighbor.

(Refer Slide Time: 56:22)



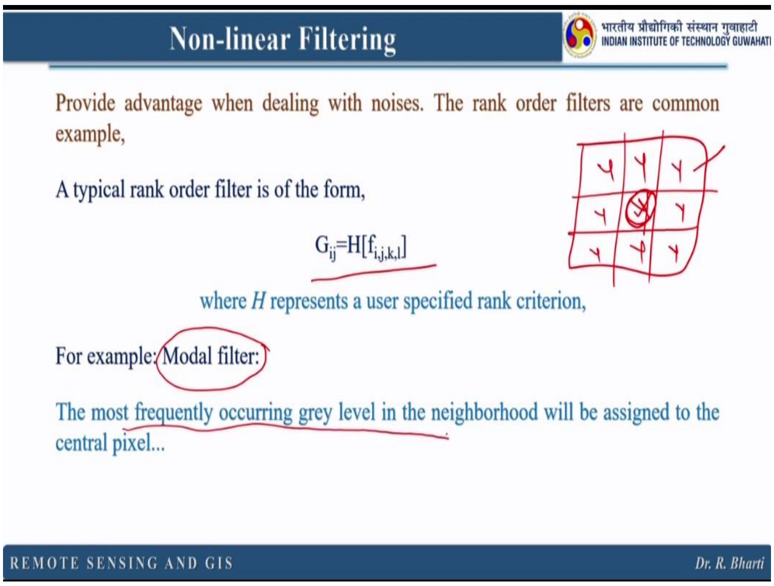
Nearest neighbor means based on the values. So, estimate the equally weighted average of k nearest neighbor whose gray levels are close to the values of central pixel right. Sort the neighbors of central pixel on the basis of similarity of grey levels. Compute the average of such k nearest neighbor and then you replace the original data or original DN value of the central pixel with the new calculated one.

So, here we have one example like this is the image pixel right and you are using 3 by 3 right and this is the central pixel for that you need to estimate the new value using this k nearest neighbor. So, how will you do that? So, the values which are close to central pixel value are these four right and why only four because your k size is 4. If you increase k=5, then you can again identify next closer value, so probably this one right.

So, in this case, when we are having k=4, these are the pixels whose gray levels or DN values or brightness values are closer to this central pixel right. So, the answer will be 39.20 is approximately 39 but if you consider k=4 right then what you will do, you will calculate (33 + 41 + 37 + 39) / 4 = some other value which is wrong. So, we have to always consider this central pixel value, then only it will be k nearest neighbor.

So, if you consider only this, this value will not come right. So, remember we are going to calculate or estimate new value for the central pixel where central pixel value presence is most important. You cannot ignore that does not matter what is your k size but you have to have this value in your calculation right. So, when you are writing a code in Matlab, C or any other programming language, you have to write that if k=4 then find the similar value from the central pixel and then while averaging, you have to include central pixel right.

(Refer Slide Time: 59:19)

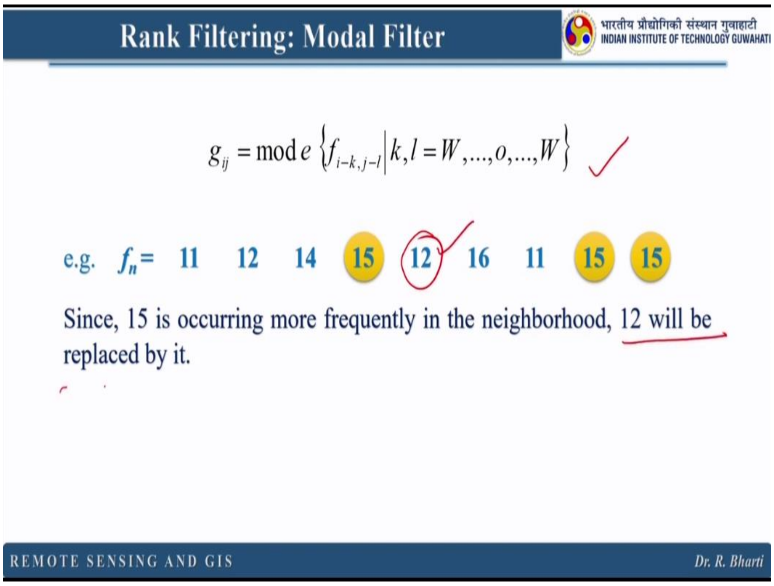


Non-linear filtering; so in non-linear filtering what happens, it provides advantage when dealing with noise. The rank order filters are common example. So, on some basis we are doing the ranking right. So, we will see. A typical rank order filter is of the form like this is the form right where H represents a user specified rank criterion for example modal filter. So, you must be familiar with all the basic statistics like mean, median, mode.

So, here what we are doing, we are going to apply this modal filter. So, the most frequently occurring grey levels in the neighborhood will be assigned to the central pixel. So, how do we do this? Let us assume that you have a very small image of 3 by 3. This is the neighborhood size and here the most occurring grey level right, so does not matter whether your central pixel is X and all are Y right.

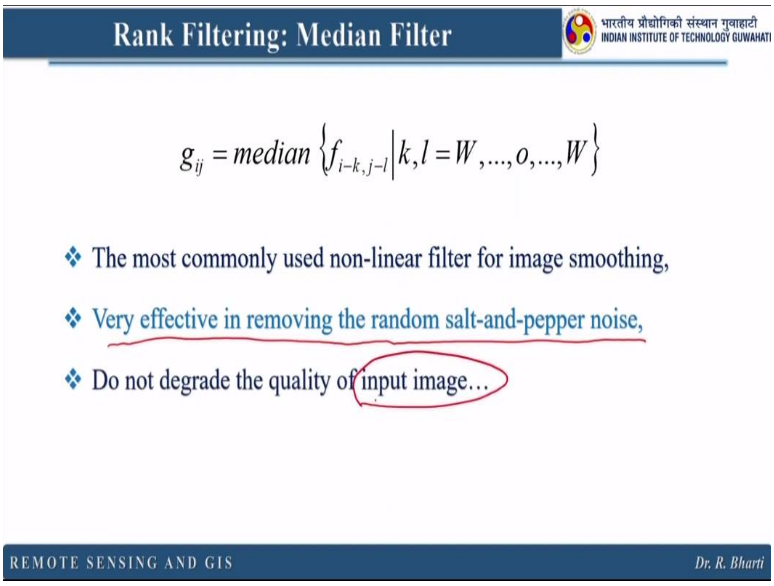
Then, what will happen, the Y is most occurring element in this particular matrix, so Y will be assigned here. So, it does not preserve your central pixel value right. It will simply replace with the ranking criteria. So, in this case, when you are using modal filter, the most frequently occurring grey level pixel value will be replaced right.

(Refer Slide Time: 01:01:05)



This is the general expression of mode and here this is one example where you have all this DN values. Now, how you will identify most occurring, you will find out what are the DN values which are coming frequently and here in this case 15 is most occurring right. So, 12 is your target, let us assume that 12 was your target, so 12 will be replaced by 15 right.

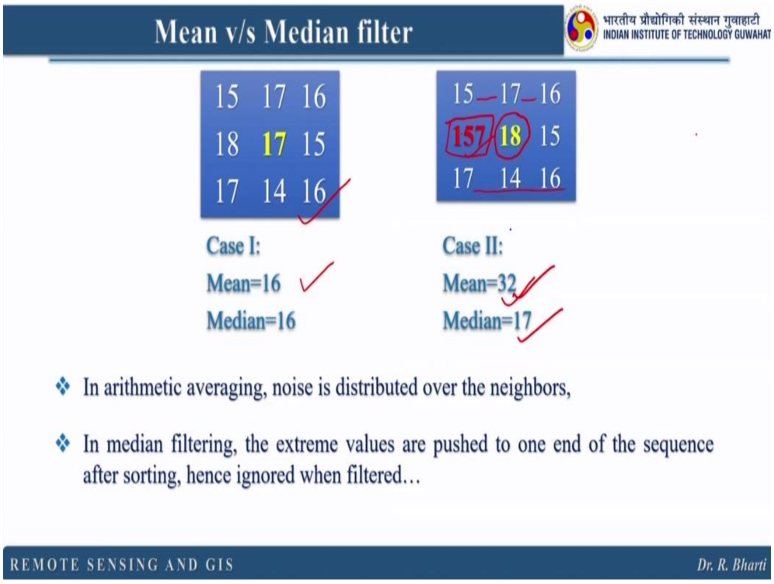
(Refer Slide Time: 01:01:36)



Median filter; it is the most commonly used non-linear filter for image smoothing. So, here it is very effective in removing the random salt-and-pepper noise, again the same thing I have explained you in previous few slides right. So, salt and pepper is basically your image will appear like you have put it some salt and pepper in white paper, in an image, printed image you put some salt and pepper.

So, that will appear and that effect will be removed if you are using this median filter right. So, here the input image quality will not be degraded much.

(Refer Slide Time: 01:02:19)

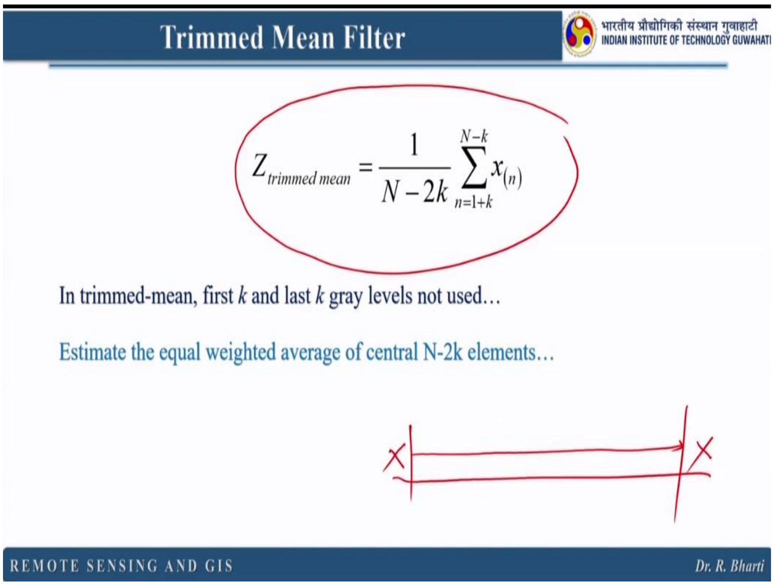


So, let us see mean versus median, which one is better. So, in the first case, when you are having this particular matrix right. So, here mean is 16, median is also 16, so here you do not have any problem which method you apply either if you use mean filter or median filter, both will give you similar result but in the second case where the mean is 32 and median is 17, so that means here if you use mean filter, your central pixel value will be changed.

Why? Because look at this particular value. So, all the other values 15, 17, 16 this is in the range of 20 right but here this one pixel value has come 157. That means there is some problem with your image which you have not corrected but here if you use mean filter what will happen, the central pixel value will be replaced with 32 that means it is not correct. So, it is always better to go for the median filter instead of going for the simple mean filter.

In arithmetic averaging, noise is distributed over the neighbors right. In median filter, the extreme values are pushed to one end of the sequence after sorting, hence ignored when filtered.

(Refer Slide Time: 01:03:59)



So, here we have seen that median filter is better than your mean filter right but what if you have to use mean filter only, then you can use trimmed mean filter, where what we do, in trimmed mean filter first k and last k gray levels are not used right and estimate the equal weighted average of central and -2k elements. So, what happens, when you are having error either it will be in the beginning or it will be at the end right.

And you will arrange them in increasing or decreasing order and then you will remove those two tails and then you will calculate the mean value right. So, this will also solve your this problem which is here right. So, this is all for today.