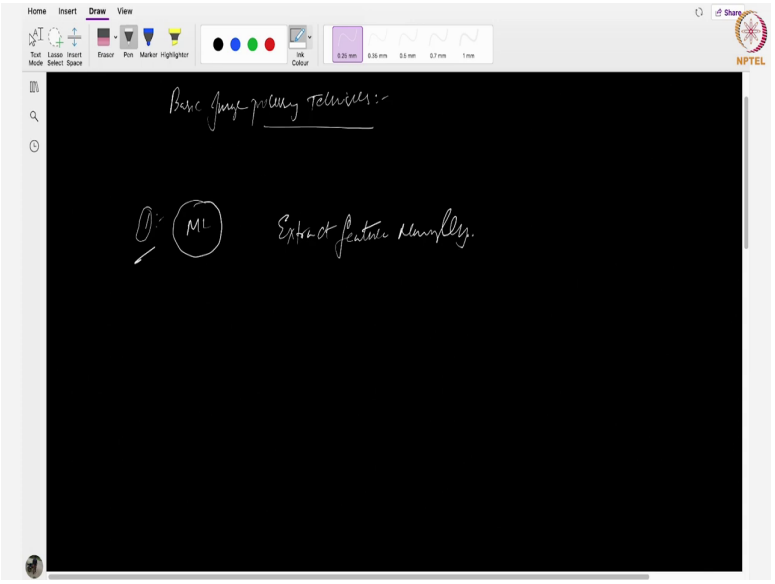
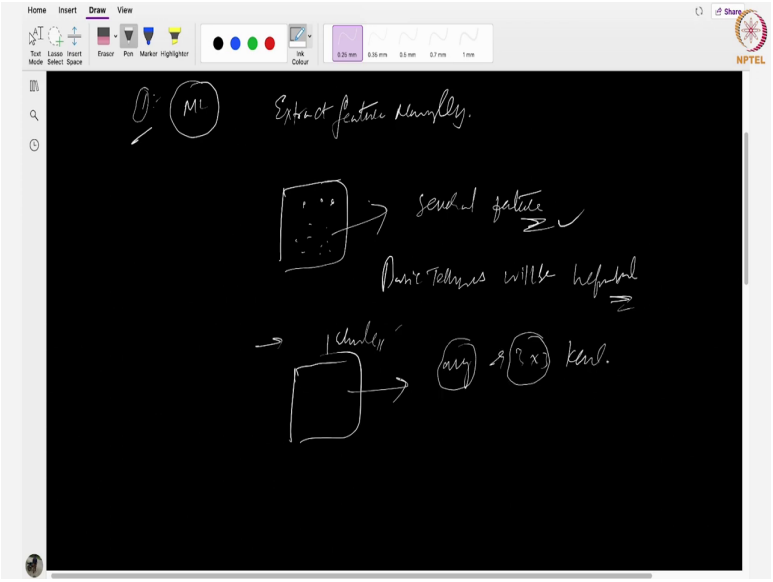
Medical Image Analysis
Professor Ganapathy Krishnamurthi
Biomedical Engineering Design
Indian Institute of Technology, Madras
Lecture 27
Basic Image Processing Technique Using Python

(Refer Slide Time: 00:15)



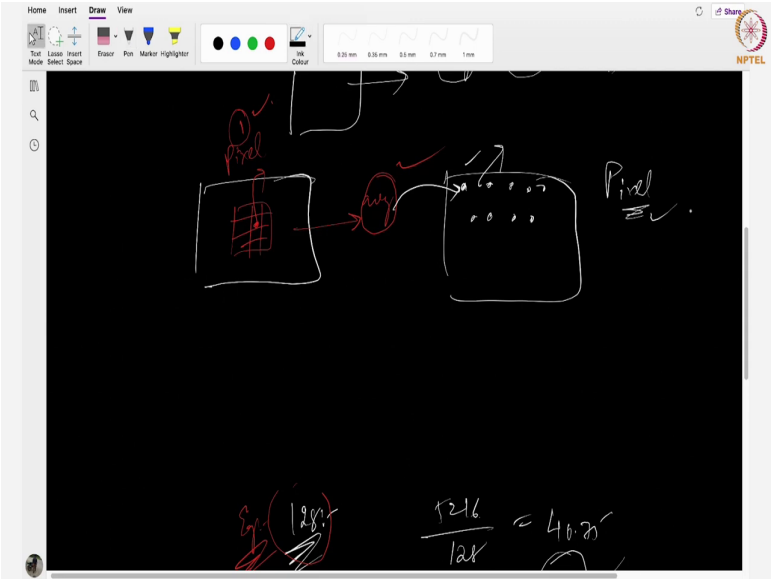
Today we are going to discuss about basic image processing techniques. Before going to show you the code implementation, let me tell you brief discussion where exactly these image processing techniques are helpful. In traditional ML algorithms or machine learning algorithms in order to feed this data, we have to extract features manually.

(Refer Slide Time: 00:49)



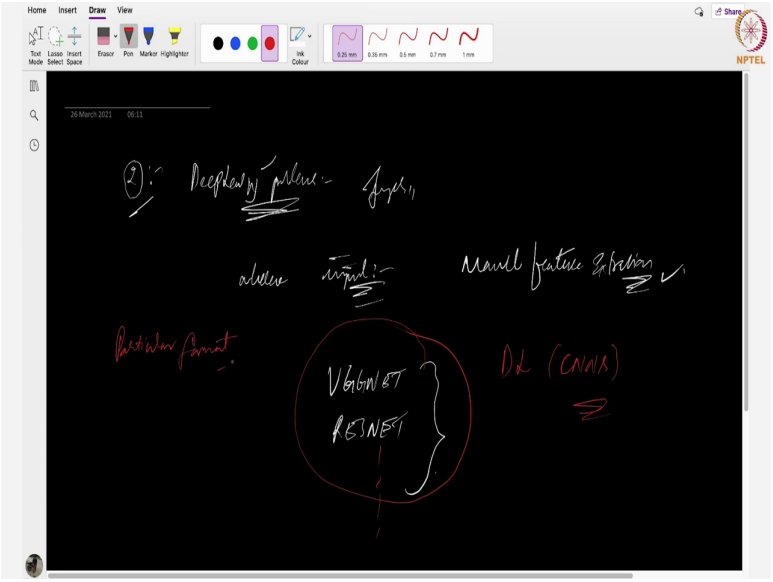
It means someone should know the domain knowledge and they have to extract the features from the images. Suppose, in our image dataset an image contains several features, to extract those features these basic image processing techniques will be helpful. Now coming back ,maybe you want to extract each pixel or in other way you just want to take the average values. Suppose I have an image. This is only one channel image. And I want to take the average values of each and every pixel for a 3 by 3 kernel.

(Refer Slide Time: 01:39)



That means, from this original image I will keep a 3 cross 3 kernel like this at a particular single pixel pixel 1. Now, I will find average value so that I will get another image where each pixel is obtained by this particular average value. Now, what happens every pixel has a new feature every pixel of this particular image has a new feature. All these things will come under image processing techniques only this is where the importance of image processing techniques will come in machine learning problems.

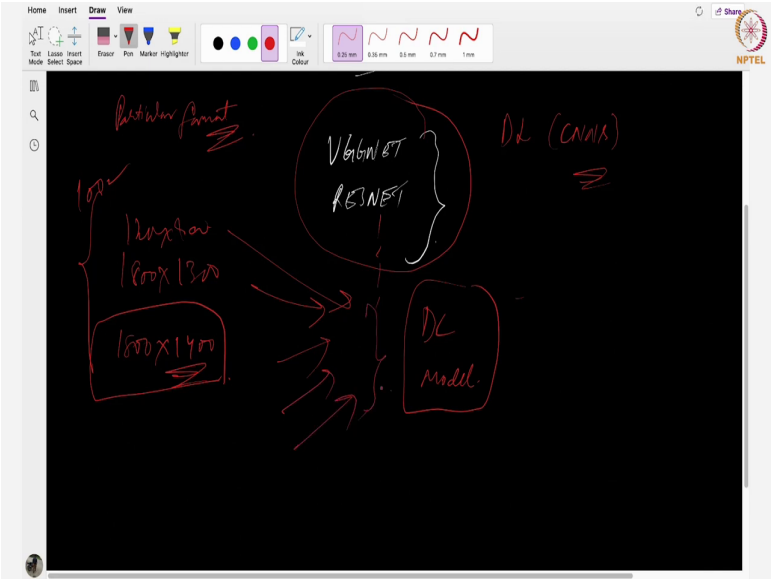
(Refer Slide Time: 02:35)



Now coming back to deep learning problems, especially for on images, images where suppose in deep learning there is nothing like manual feature extraction, features are extracted when training goes on by default by the model itself. So, what are the requirement of image processing techniques here?

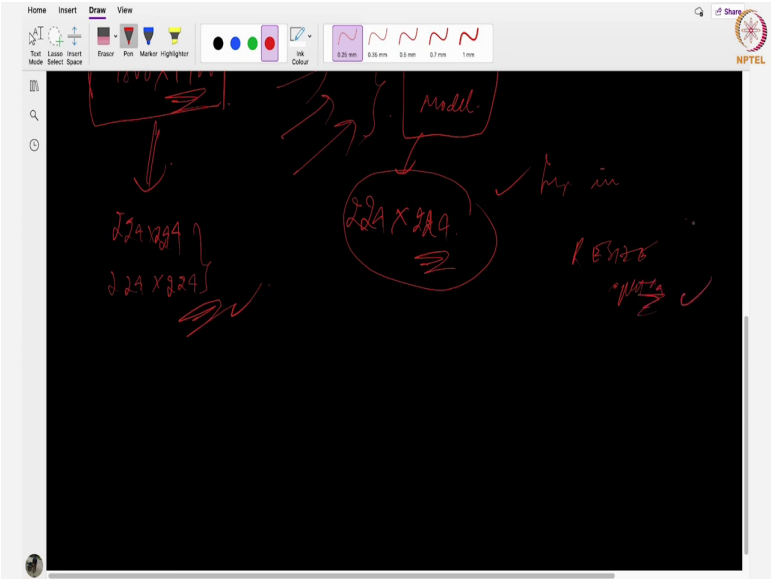
Suppose, let us take some VGGNET model or RESNET model. Maybe you do not know what these are. But not necessary. Just imagine that there are some models of deep learning, especially in CNNs models. Likewise, several models will be there. Now, how the expected data means image data means they want image data to be in a particular format they want image data to be in a particular format.

(Refer Slide Time: 03:43)



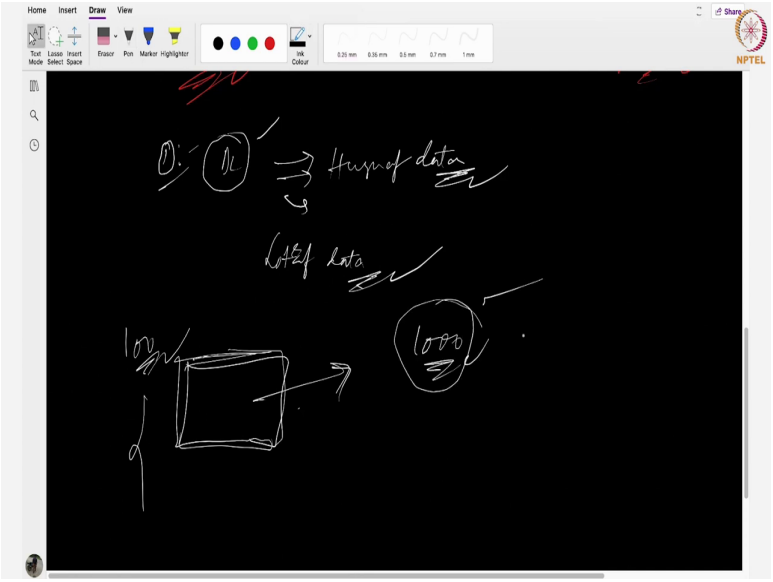
Suppose, I have images some around some 1800 cross 1400 this is the size of the each and every image or some 1800 cross 1300 or 1200 cross 400 maybe suppose I have some 100 images in a particular data set and each and every image contains consists of these sort of dimension. They have unique dimensions. But at the end of the day if you want to feed these kinds of entire data into particular DL model, every image should be in a particular size only.

(Refer Slide Time: 04:27)



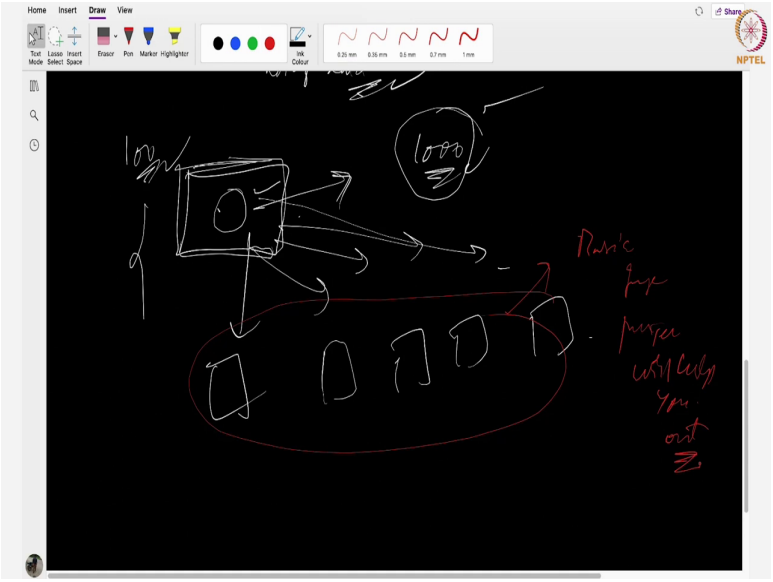
Suppose, this model accepts input in the form of 224 cross 224, so what happens? Before feeding this entire image data or each and every image into this particular model you just have to resize each and every image in this particular entire training data to the size of 224 cross 224. This is one such case of necessity of our requirement of image processing technique that is nothing but resize operation. Now coming back to in one more case we will I will tell you.

(Refer Slide Time: 05:10)



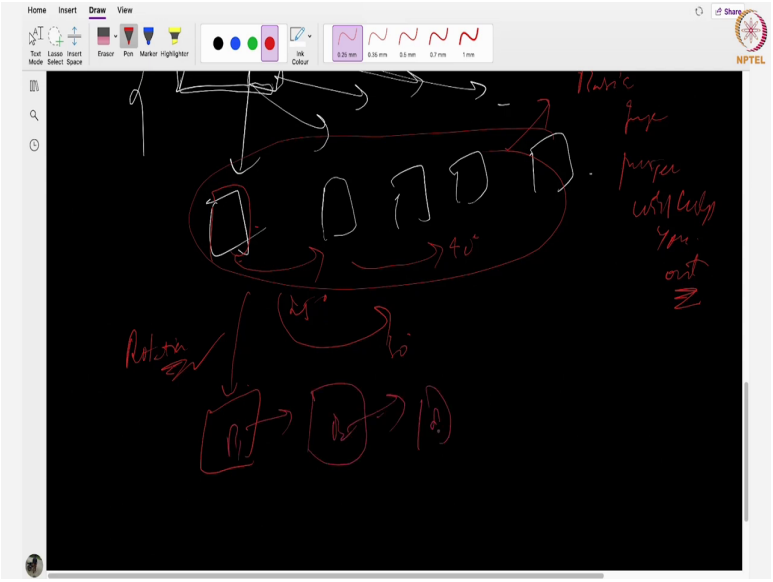
Suppose in DL model DL models are very much hungry of data. In the sense, in order that the model to be performed well they need lot of data lots of lots of data but we have only one such sort of data. Suppose, I have 1 image suppose maybe you have some 100 images in a particular given training data set but I want to increase this entire data to some 1000 data sets.

(Refer Slide Time: 05:54)



So, what should I do? I will take a single image and I will do several image processing operations on this particular image alone and I will generate new images out of it. At that point of time also basic image processing techniques will help you.

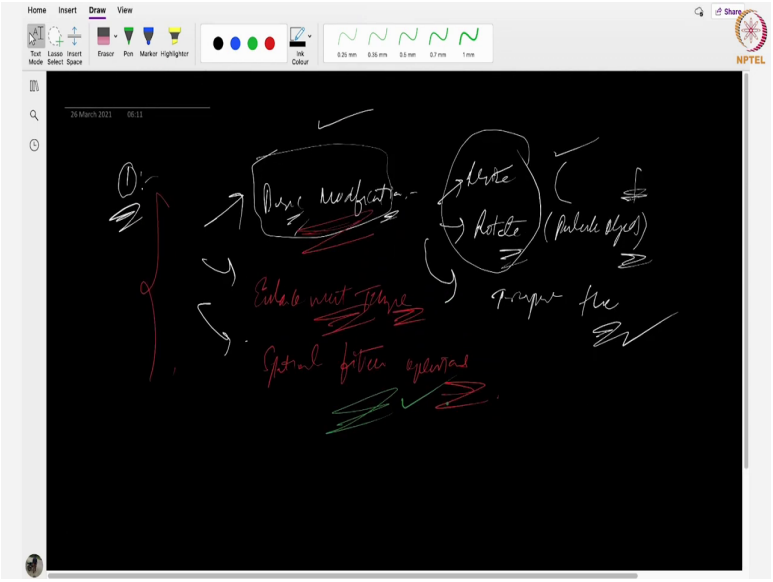
(Refer Slide Time: 06:20)



One such operation is maybe I have rotated each and every image suppose 1 image is like this and I want to rotate the image to 25 degrees some 30 degrees like 40 degrees likewise one such operation is rotation and the same image I want to increase the brightness up to some other value brightness 1, brightness 2, brightness 3.

Or also I will I want to do some special operation special filter operation. Likewise in summary I can say that in machine learning and deep learning problems basic image processing techniques are very much helpful in order that the model to be performed well. That is the whole summary.

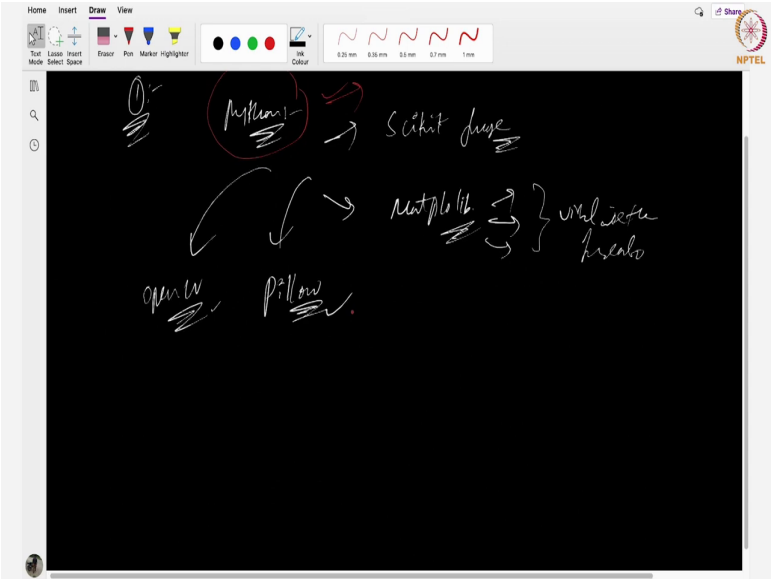
(Refer Slide Time: 07:09)



The entire lecture has been divided into 3 parts, first one is some basic modifications. Basic modifications like resizing operation suppose image is having large size I want to reduce it to some smaller sizes. Second thing is rotate; I want to rotate the image to a particular degree to a particular degree. And one more thing is I want to transpose the image another image is nothing but a matrix only transpose version of a matrix in another way.

All these operations come under basic modifications of image processing. Second thing is enhancement techniques. Third thing is spatial filter operations. In this lecture I will discuss 3 things, first one is basic modifications, second thing is enhancement operation and the last thing is spatial filter operation on the given images. I will show you through the code how exactly to do it later on.

(Refer Slide Time: 08:36)



Before that let me give a brief introduction, usually in python we have several image processing libraries are available like scikit image is one such a library, second thing is Matplotlib, Matplotlib is for even to extract some information from the image at the same time to visualize the image also to visualize the image also. Third thing is pillow, one such library. Next thing is OpenCV. This is one such library.

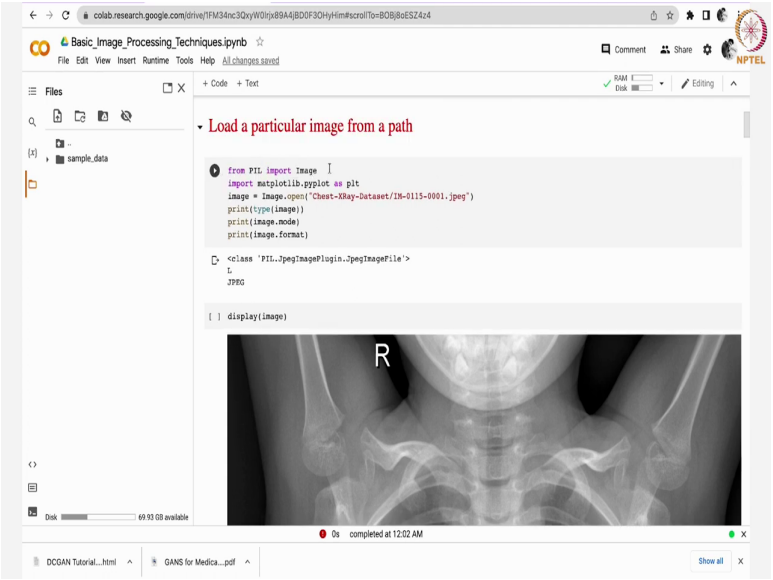
So, keep in mind that end of the day in python there are several libraries that are available for us to do basic image processing techniques. That is the final summary.

(Refer Slide Time: 09:39)

Whatsoever, but here in this lecture I mostly focus on pillow. Using pillow library let us see using pillow library let us see what sorts of operations are to be done or can be done. Open the drive google drive, see here see here here I have downloaded some chest extra data set. Here, if you observe the original image is this one. This is the original image. I have downloaded this one from the kaggle chest x-ray data set that is available just like the chest X ray images here.

This is some around some 2 GB data is available I have downloaded that data and I have started using those images to explain this entire process. Like suppose, this is what the original image looks like. So what happens is I have loaded this image and I have done some basic image processing operations and I have saved them back to these files like rotate the image, resize the image and transpose the picture, crop the picture. Let me show you.

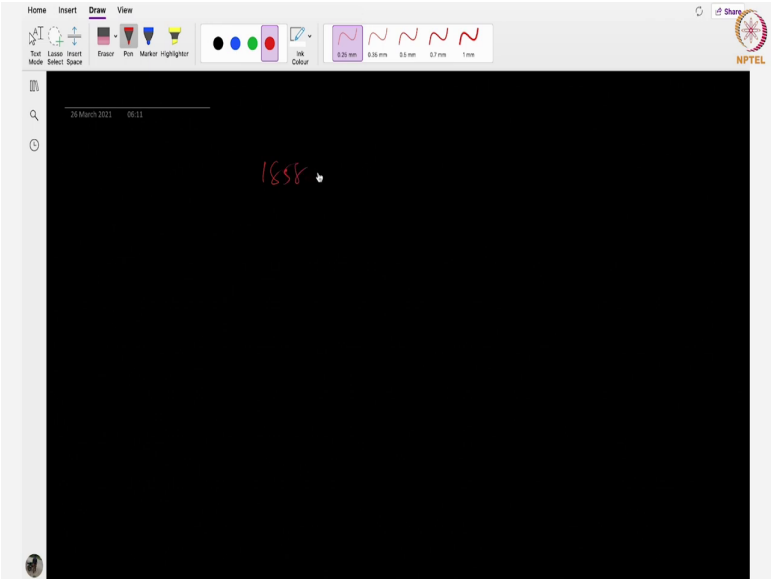
(Refer Slide Time: 11:16)



As I have already told you see here you just have to in order that to use a google colab file you just first go to go and use this mount drive google drive, later on here you just have to go through this running this code I will just explain what are the operations are available. See here we are using pillow library here from pillow or PIL we will import image later you should give a particular correct path to whichever image you load here that is very much important all the remaining steps are not that difficult.

So, what happens is I just loaded this image see here and I have shown the image here. See this is very large image, let us see in which format the image is like jpeg format here. Similarly, L indicates only one channel. The next thing is type of image. It is a image file class from the pillow library that provides image module. Just import that one image that will help you to load this image jpeg file.

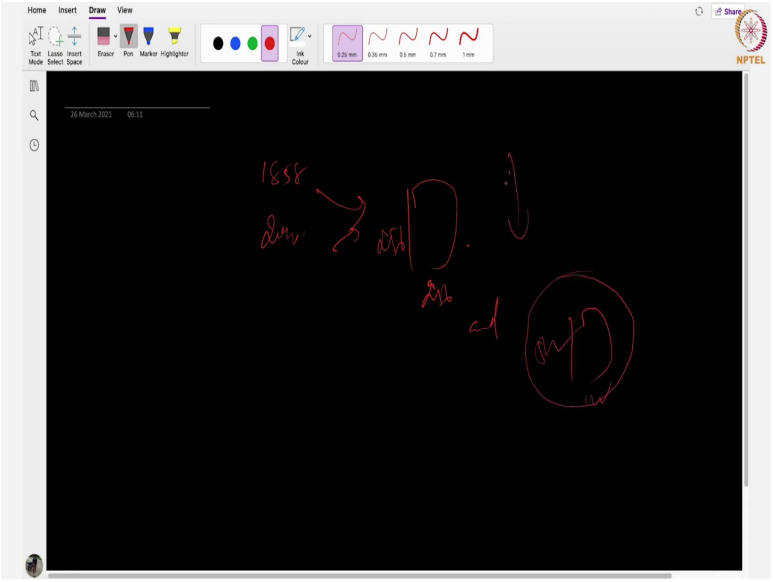
(Refer Slide Time: 12:27)



Next thing is I just want to extract what is the height and height of width of the image so that we will get to know the size of this entire image. See here width and height image size it is having some 1858 to 2090. Very huge image actually. So, usually the CNN algorithms whatever we use later on all of them will expect the input data somewhere around from 224 to 224, 512 to 512, 256 to 256 likewise.

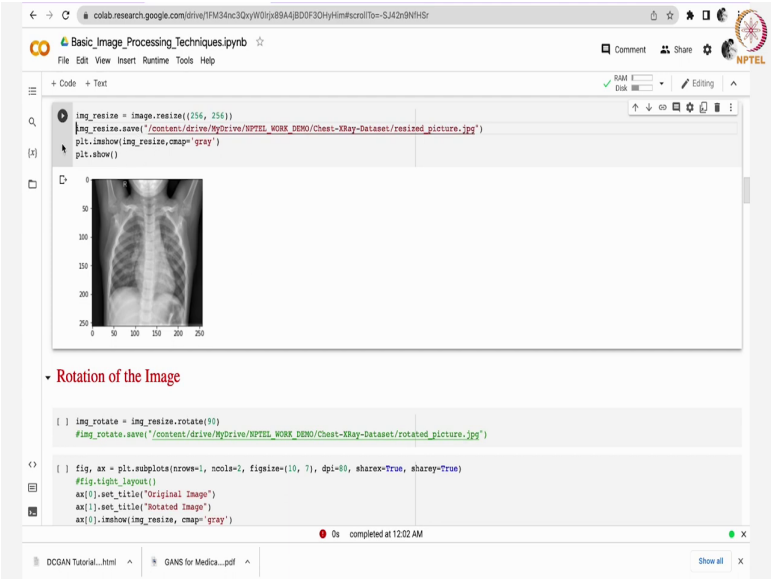
So, the first operation I will discuss is resizing operation. This image contains 1858 to 2090.

(Refer Slide Time: 13:06)



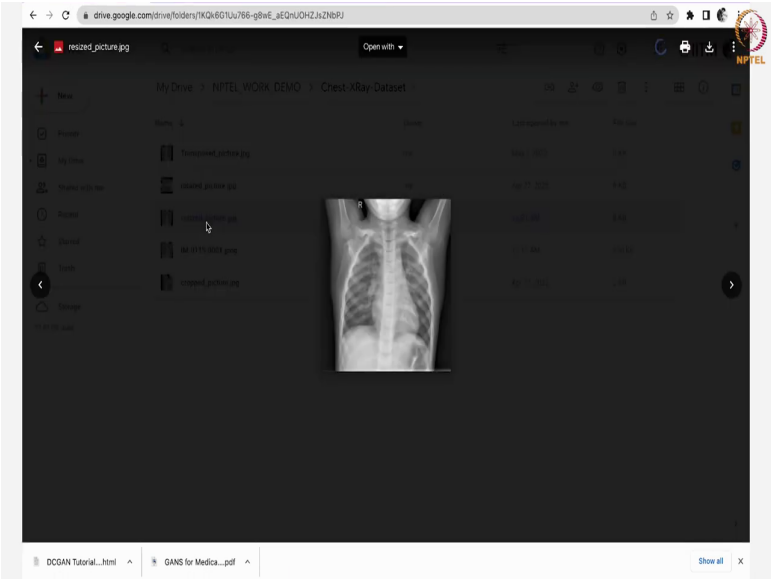
Now, I want to resize this image to some 256 cross 256, similarly 512 cross 512. This operation can be done using this code. Just go through this code ,you will get to know easily.

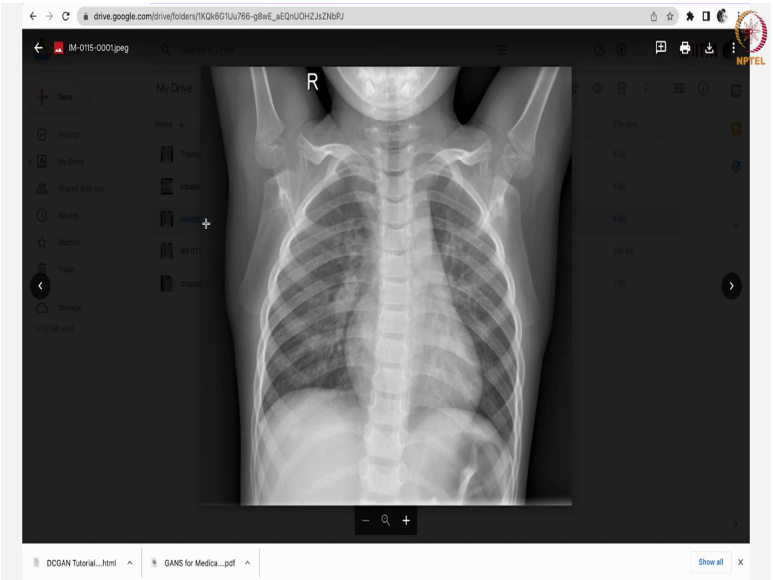
(Refer Slide Time: 13:25)



Now image has been resized, see here from this image to this image you can see here. After doing this operation you can even directly go and save this one here using this code. Actually I have commented out; just uncomment that line and you can say wherever you want to save this line giving this particular path is very important other than that everything else is simple only.

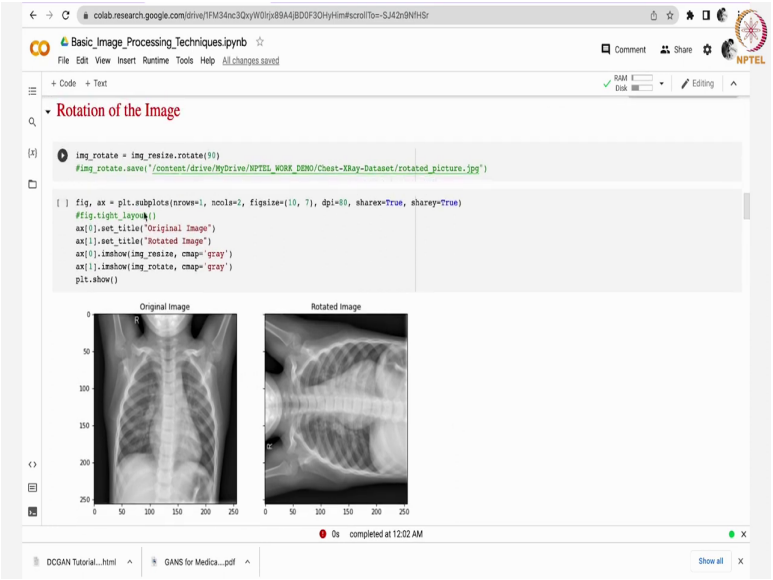
(Refer Slide Time: 13:51)

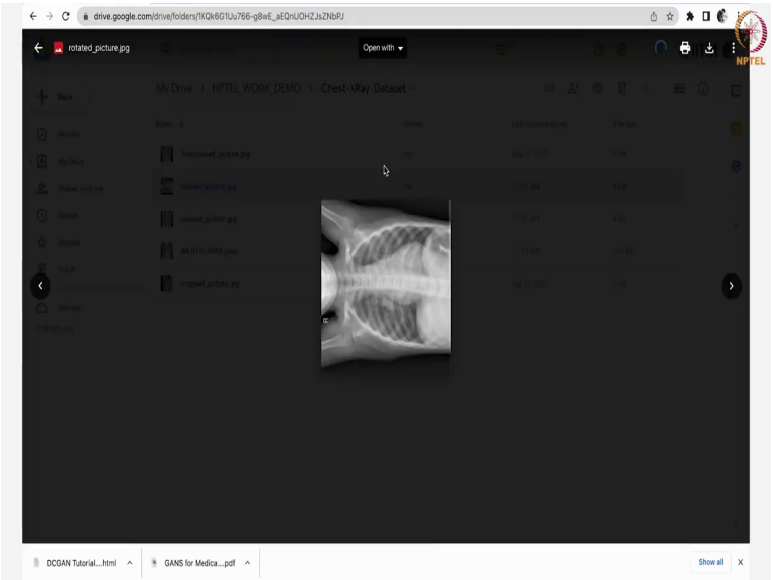




See here after saving this one you can see here resize the picture see original image is this much size this much size I have just modified it to 256 plus 256. That is one first operation.

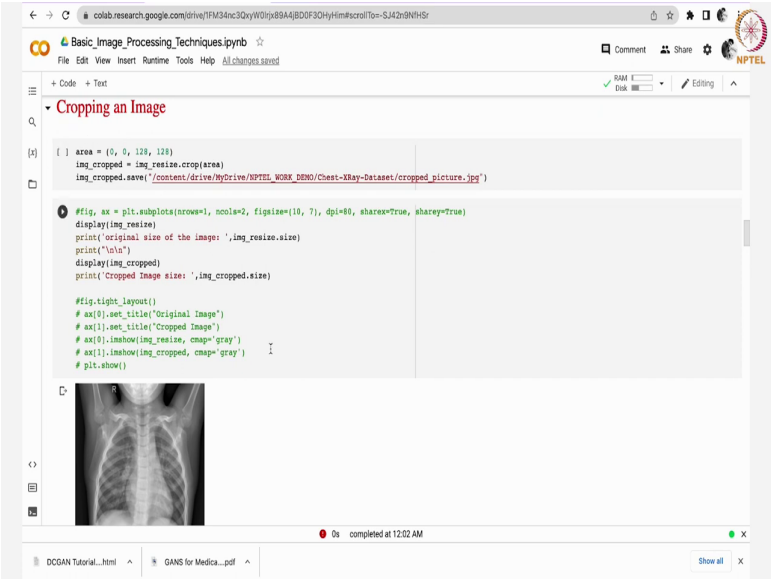
(Refer Slide Time: 14:04)

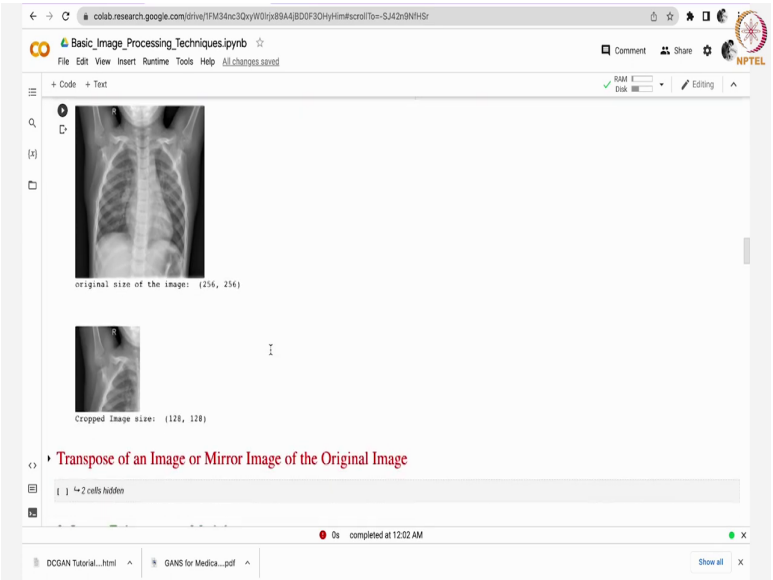




Now, let us get back to another operation rotation operation see here using image underscore resize dot rotate we are using rotate here you can directly rotate this image there. They have given something like 90 degrees see if I keep this image in 90 degrees or that will be given like this and I have saved this file also here see rotated picture. Now, that is rotation operation even you can keeps like 45 degrees or 60 degrees or 70 degrees. Whatever you want all these things will be helpful in data augmentation.

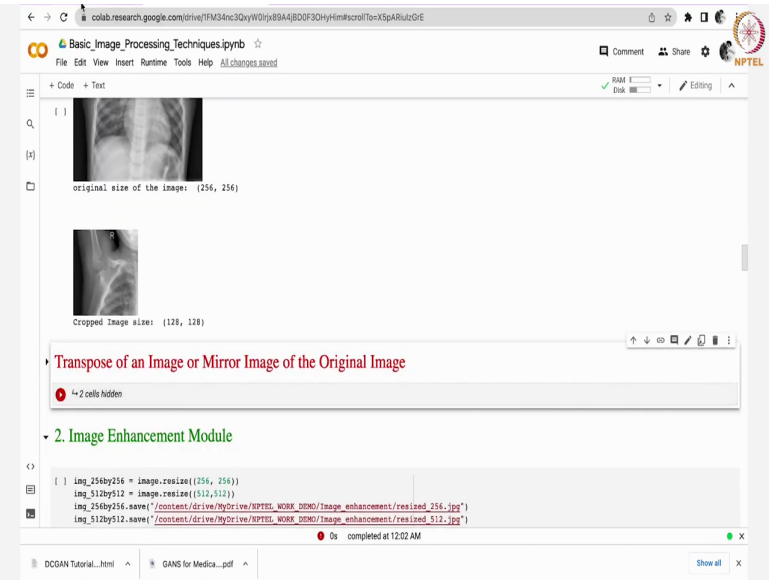
(Refer Slide Time: 14:45)

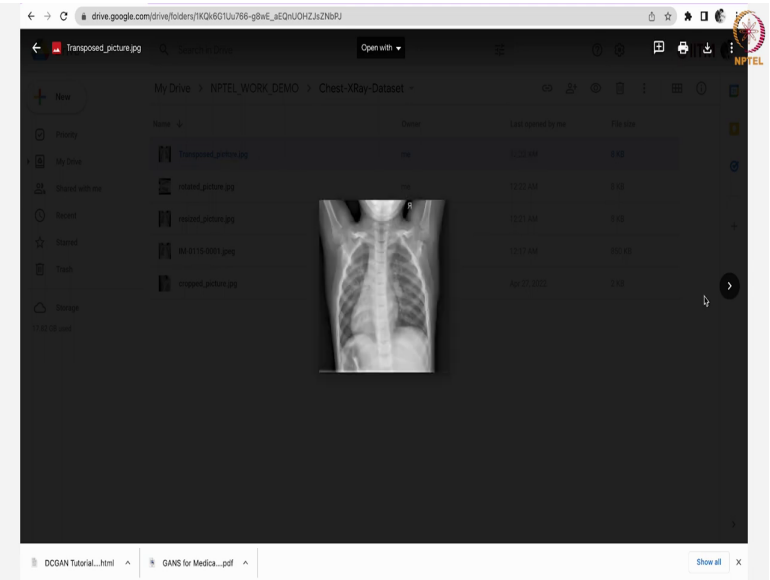




Now, coming back to cropping the image. Suppose I have this huge image of 256 to 256 length but I want to extract a particular portion from this image. At that point of time, you will get to know you will see the help of cropping an image very important. See from this entire portion I have just taken out this 128 comma 128 part from here to here and it has been saved as another. What we have seen first one is resizing second one is rotation third one is cropping.

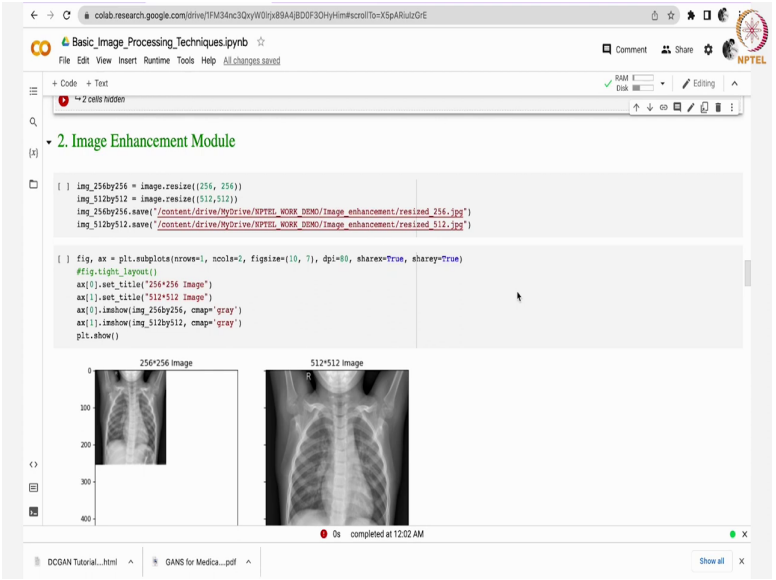
(Refer Slide Time: 15:17)

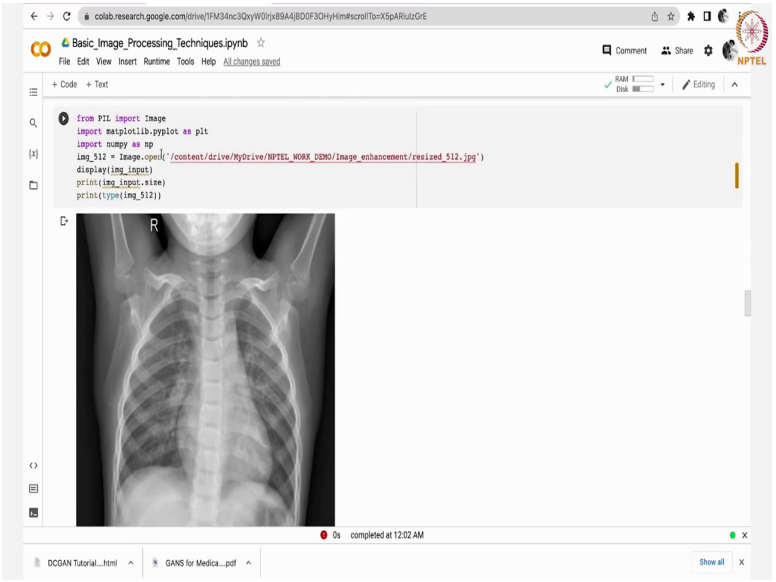


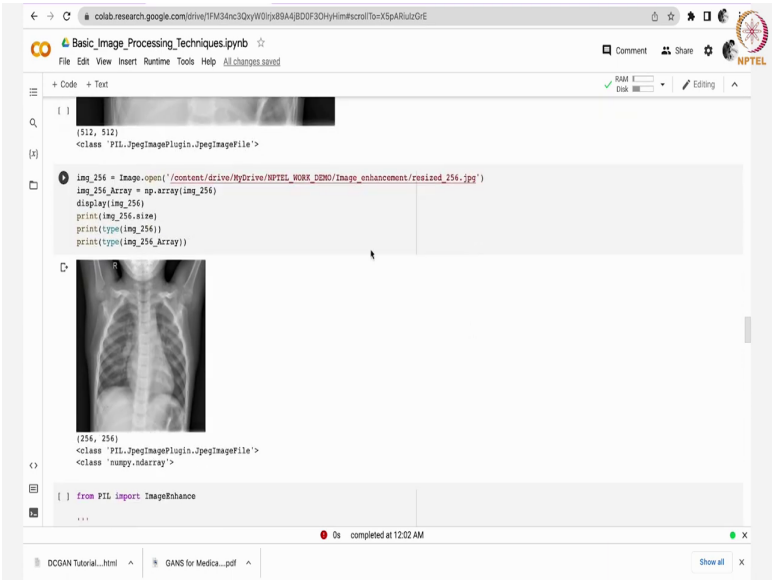


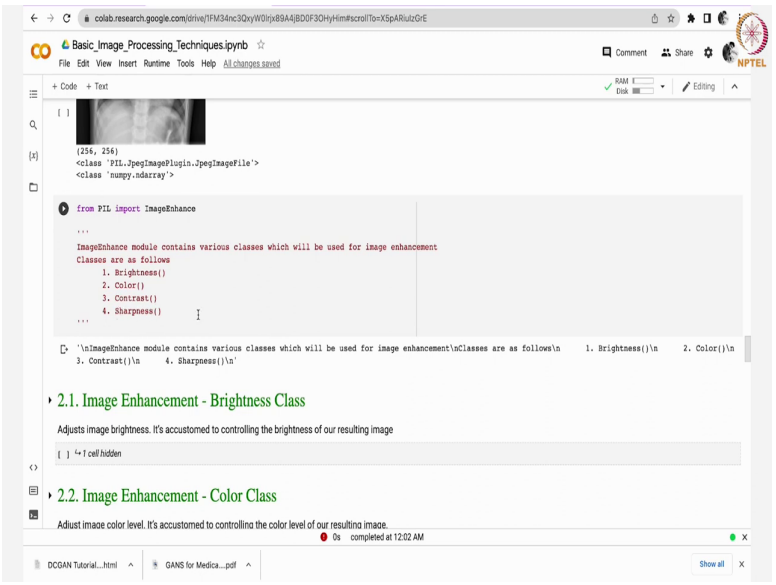
Now, let us get back to transpose of the image you can see here transposing picture this is how the transpose picture will look like. Whereas, the original picture will look like this. This is 90 degrees rotated image. These are the few basic modification operations done here.

(Refer Slide Time: 15:38)





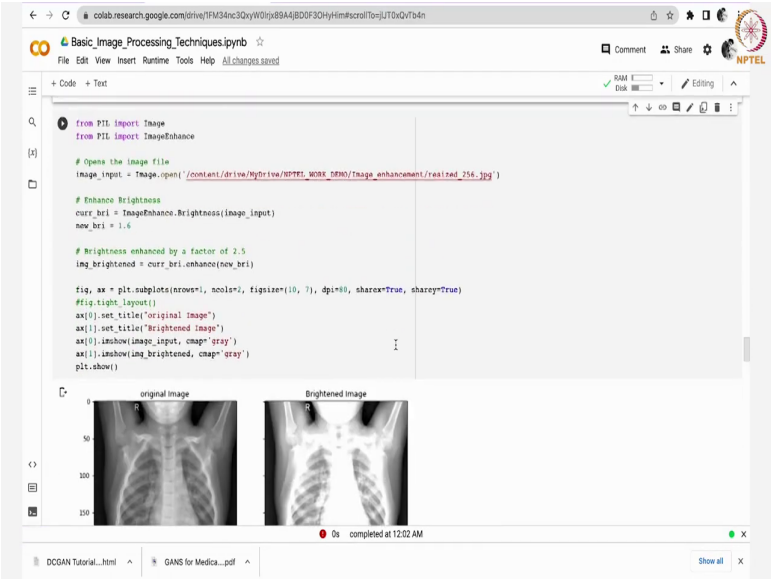


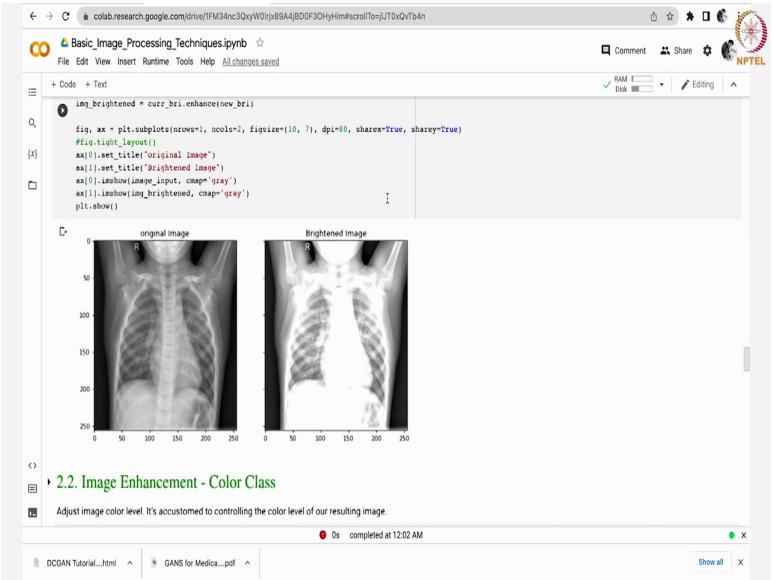


Now, coming back to now let us get back to image enhancement module. In the first case I just wish to increase the. See here two images have been loaded 256 cross 256 and 512 cross 512. In PIL image in PIL from pillow you can download, sorry import image class, later you can display here it is having a size of 512 cross 512.

And I want to enhance the image you know. Enhancement can be discussed in four ways first of all increasing the brightness another one is colour contrast and another one is sharpness. All these operations are applied on the given image.

(Refer Slide Time: 16:37)

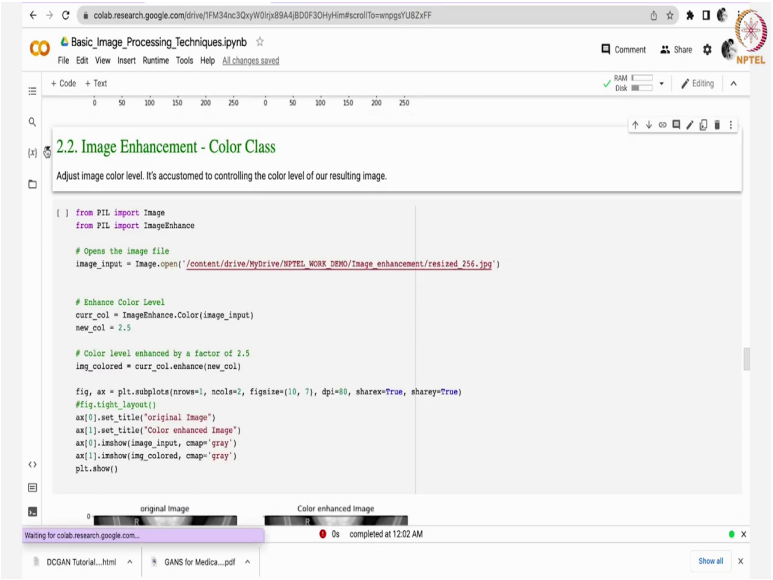


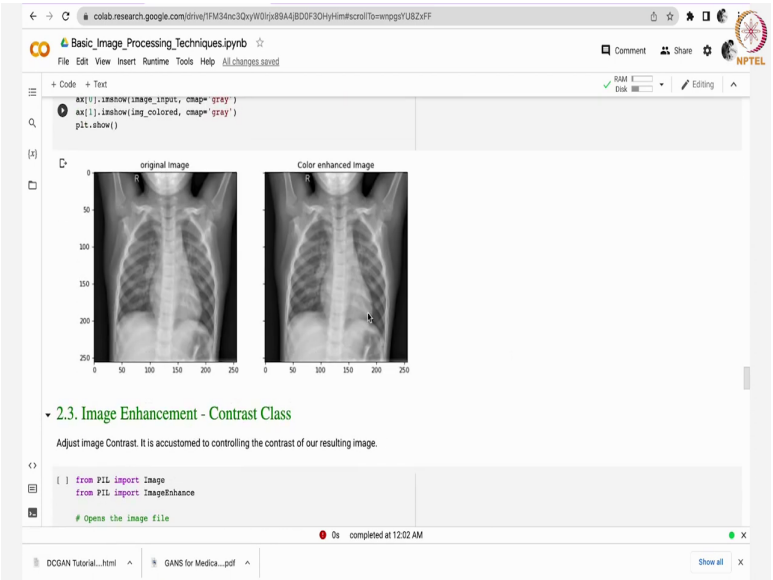


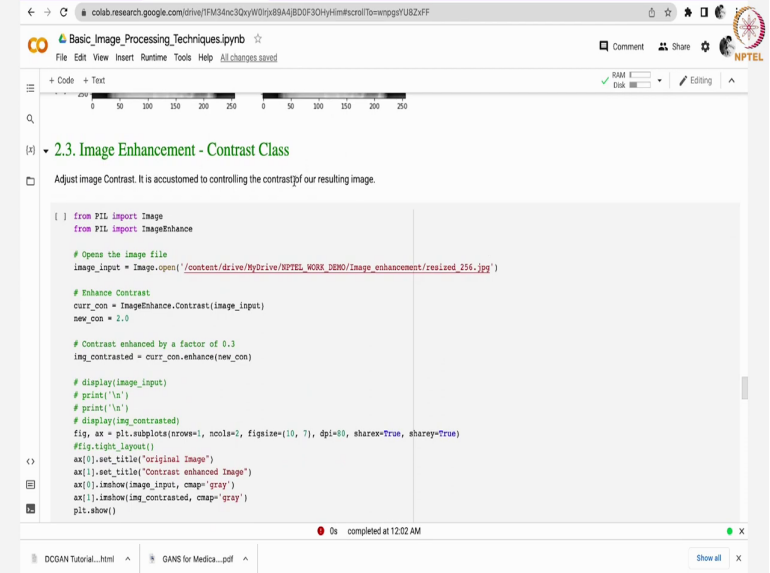
See for example, image enhancement module. Suppose this on the left side you can see the original image after increasing the brightness. This is how it is looking like. You just have to here we have something like image enhance dot image enhance module has been imported from PIL image enhance dot brightness of image input.

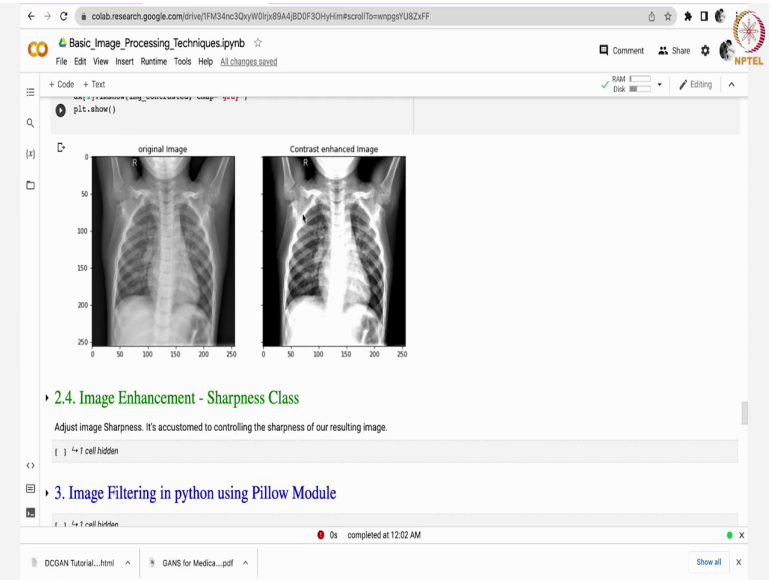
If you change this value new brightness value accordingly this particular brightness will change you can play around with this one that is not that difficult.

(Refer Slide Time: 17:11)





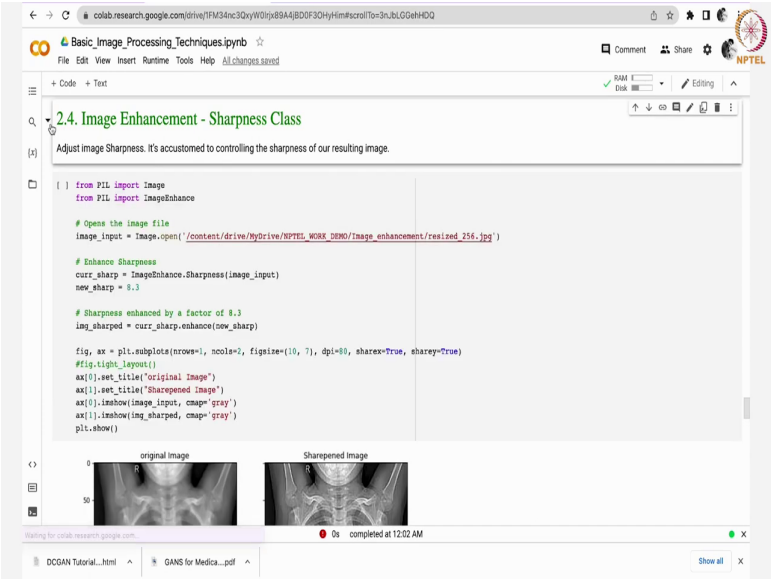


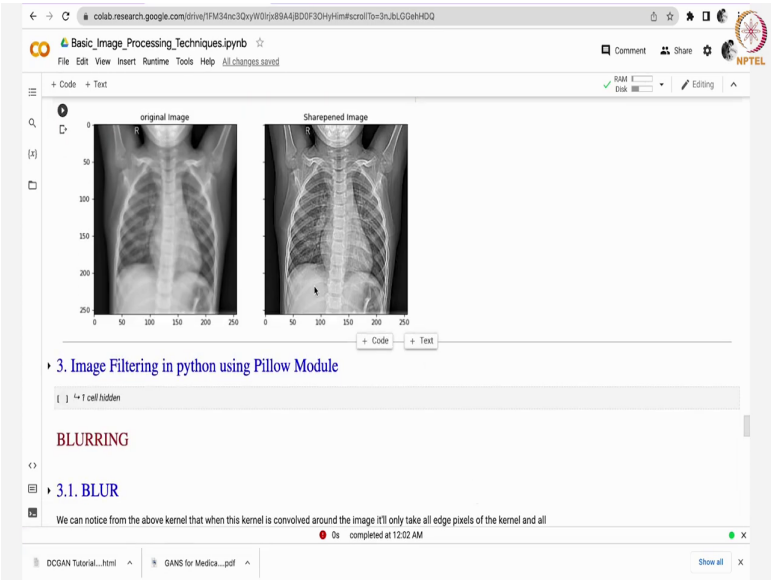


Next thing is colour class. You can see this is how the output looks like. Similarly, contrast class contrast this is another image enhancement operation here also we have something like image enhance dot contrast. See here image enhance dot colour see here image enhanced dot brightness.

Colour class, contrast class see here original image look like this in the left side. This is how contrast enhanced image looks like in the right side.

(Refer Slide Time: 17:48)

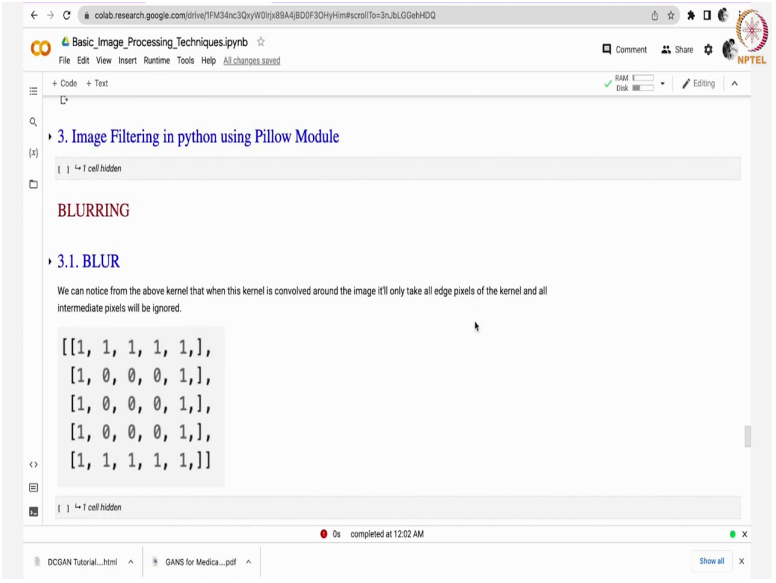


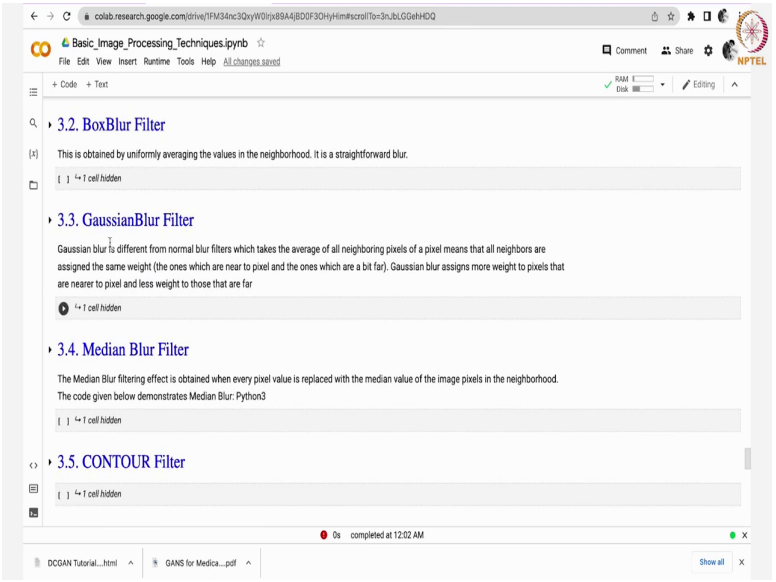


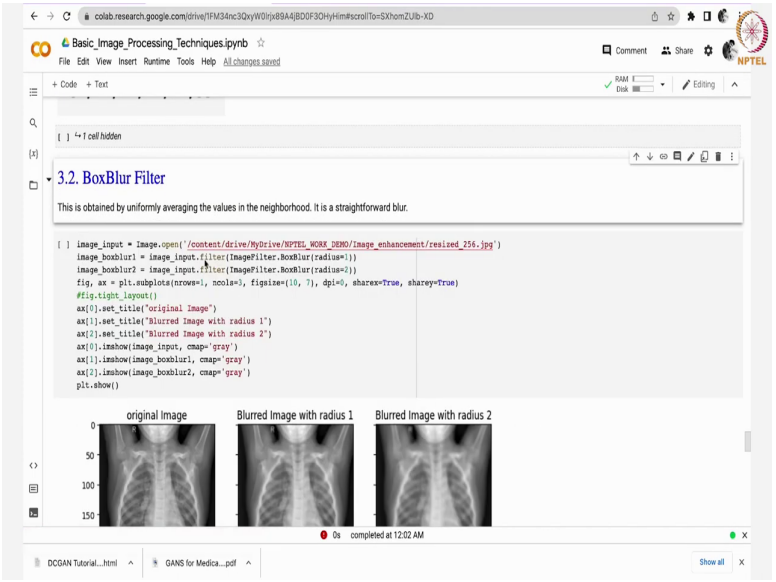
Next coming back to sharpness class, see how the output looks like. See this is lower left side image original image and say right side image is sharpened image. This is how the image enhancement operation works. Pillow provides four classes. In summary I can say that brightness class, colour class, contrast class and sharpness class.

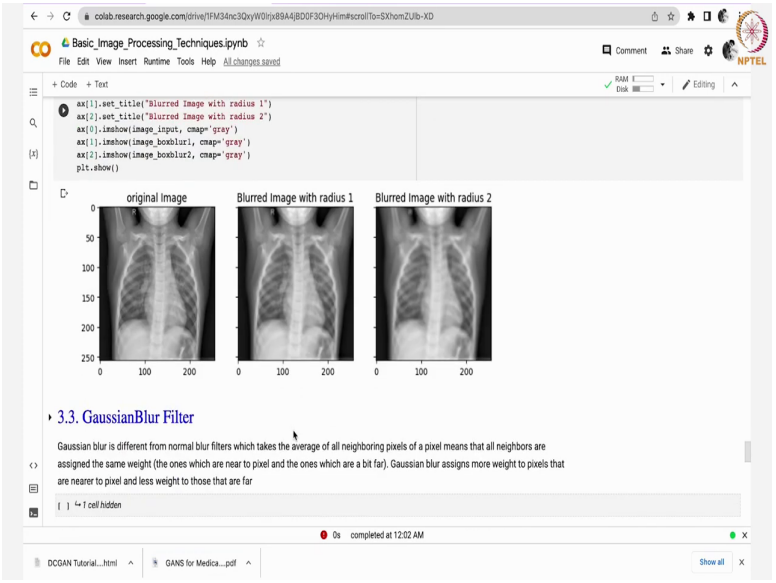
Based on your requirement just apply this code on to the data whatever data you have so that you will see these details: sharpness colour and contrast enhanced, colour class and brightness class that is it nothing else. All these operations come under image enhancement operations.

(Refer Slide Time: 18:37)





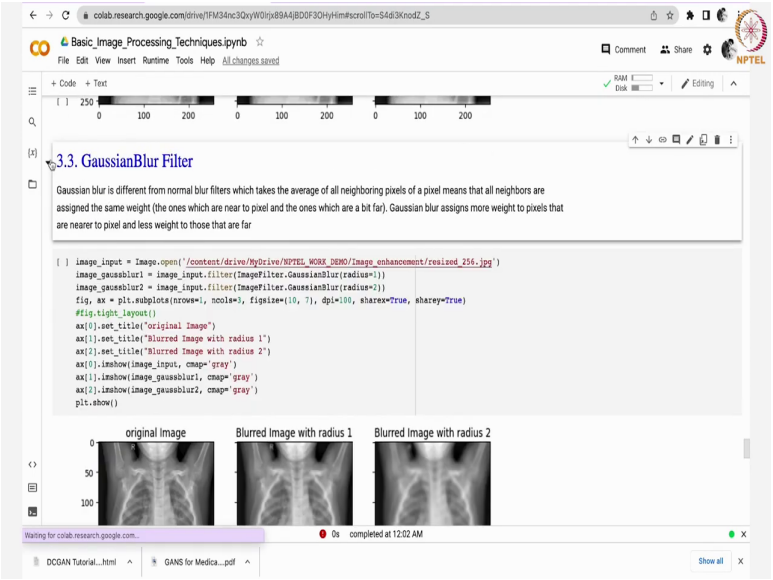


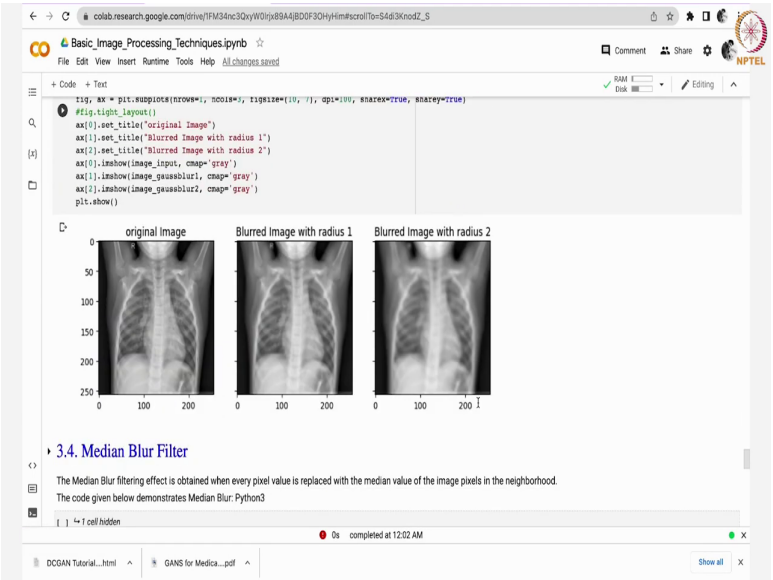


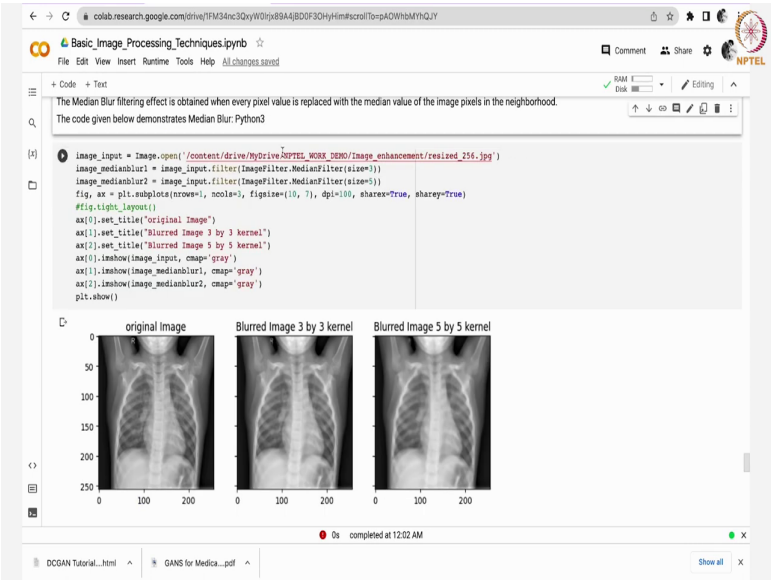
Now, let us get back to spatial filter operation. Image filtering in python using pillow module. Suppose, I want to blur the image it can be done it can be done using blur class, box blur filter similarly gaussian blur filter and medium blur filter. Just check out how the outputs looks like in each and every blur.

See here: original images look like this with the radius of one this is how it has blurred with the radius of two it has the blur operation has been increased more. See here the filter that is used is this one maybe the image size may be different here we have taken some 256 plus 256. On this we are using here 5 cross 5 kernel this is how the output looks like.

(Refer Slide Time: 19:40)

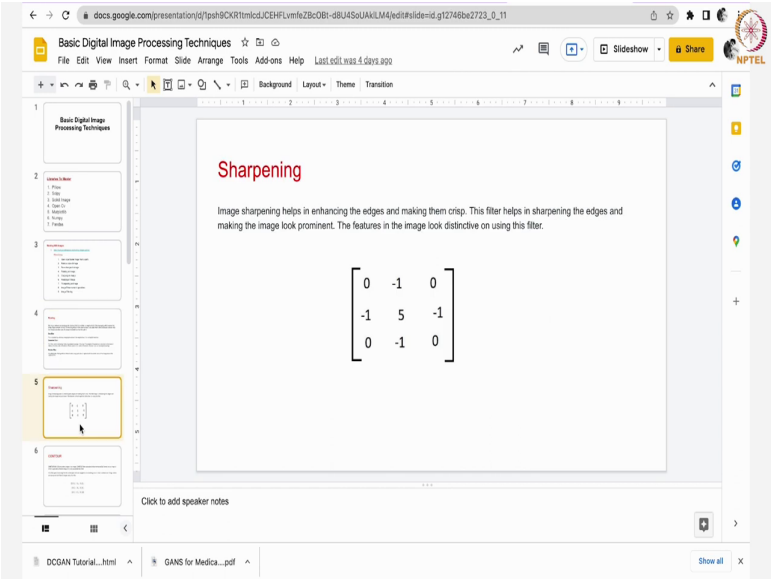


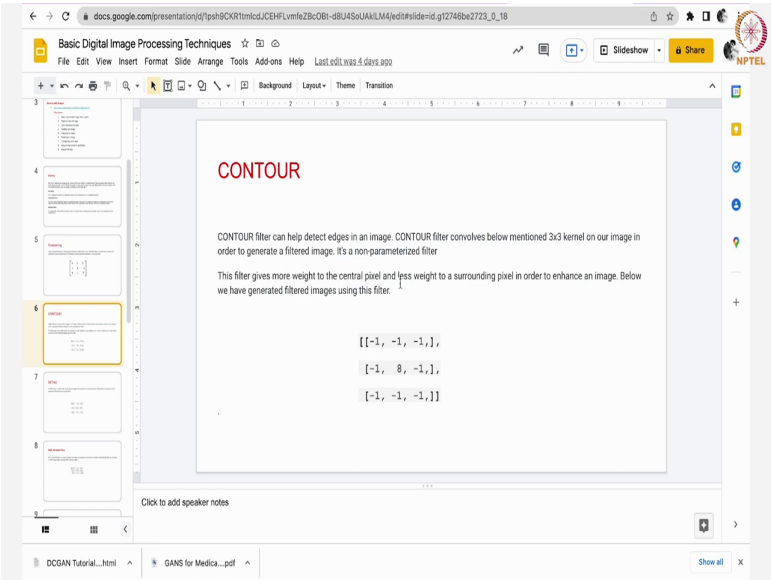




Next thing is suppose, if we apply gaussian blur filter this is how it look like. See with radius 1 it is output looks like this with the radius 2 output looks like this now get back to medium blur filter check out here this is medium blur gaussian blur this one is box blur and another one is black. Just you apply this code it will give you for this for your particular data that is it there is nothing else to discuss more.

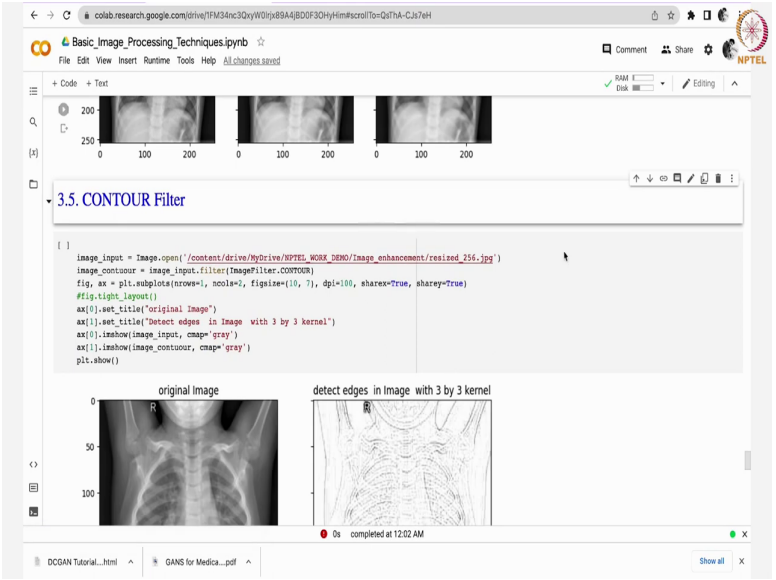
(Refer Slide Time: 20:19)

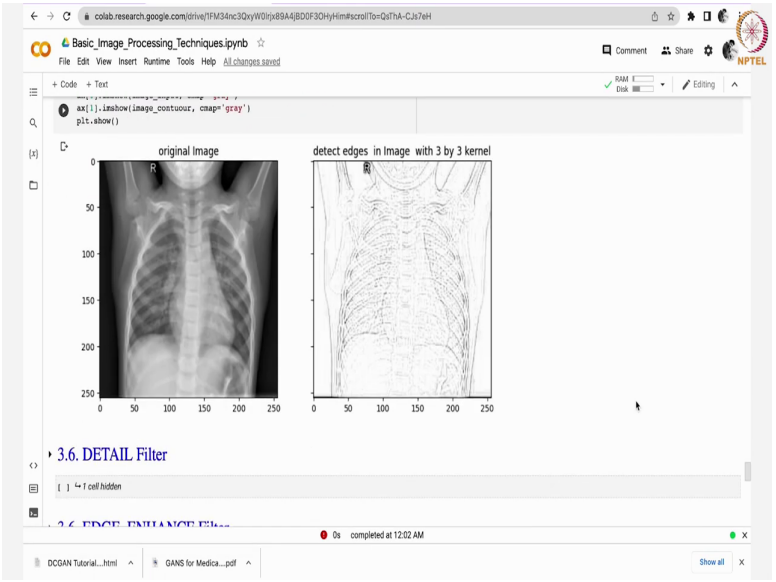


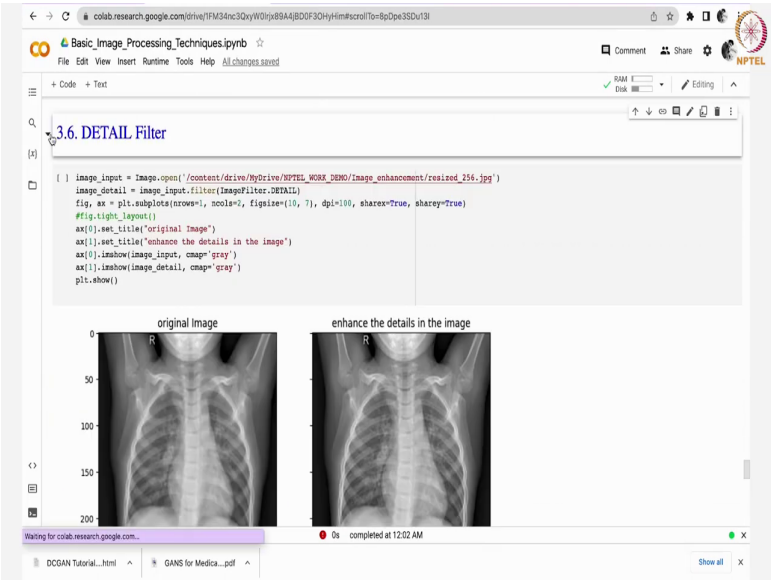


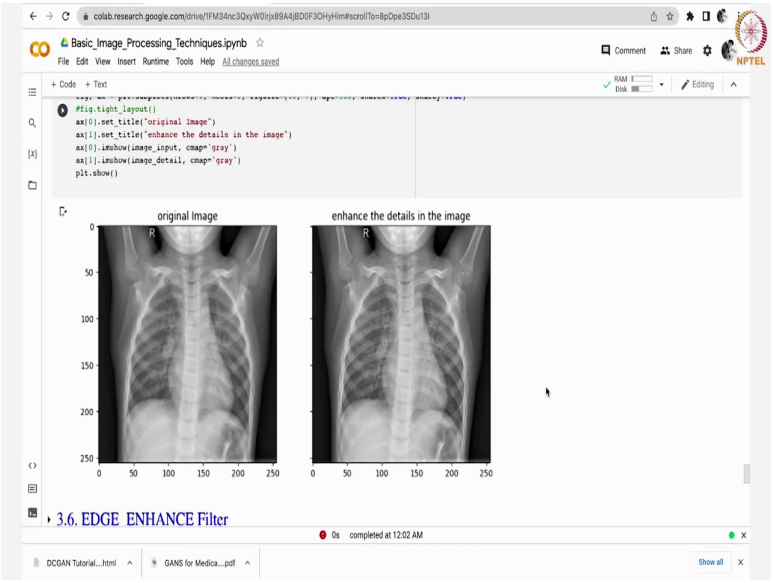
This is for sharpening. You can keep this filter value automatically can be applied. Now, coming back to contour. It will help you to detect the edges in an image let us see how it helps.

(Refer Slide Time: 20:35)



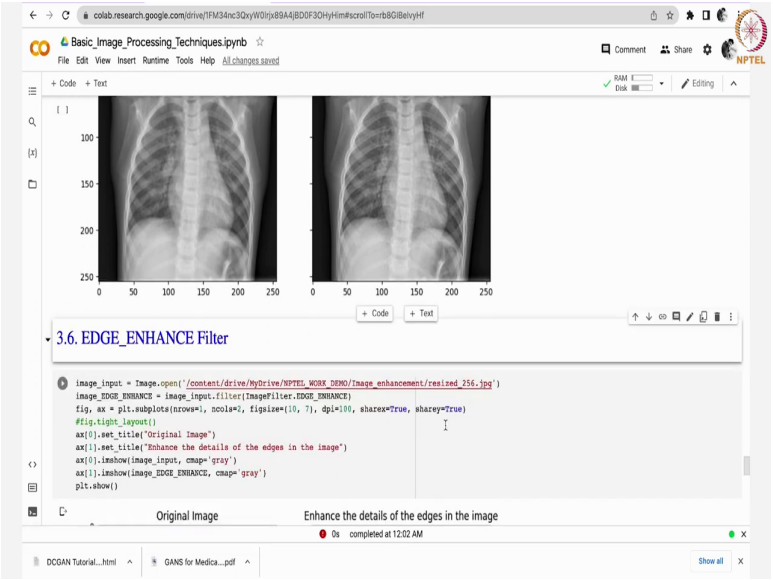


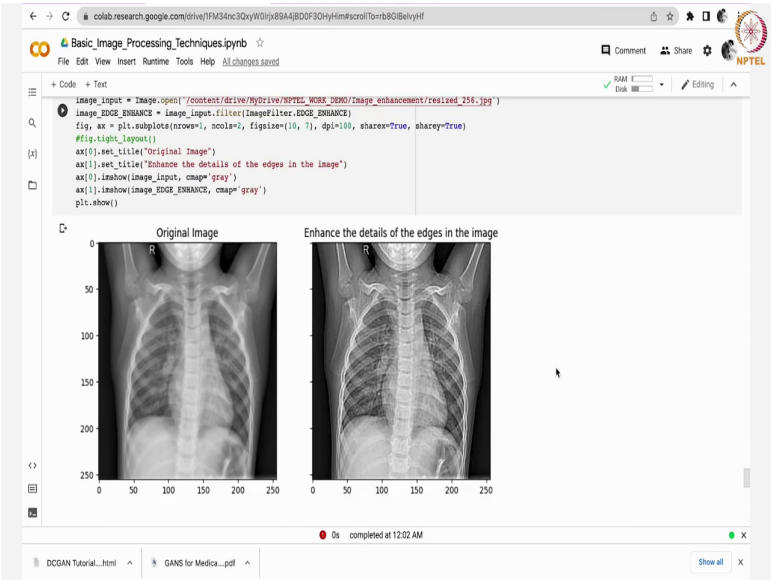




See here the code contour just see it detects the edges this is the input image and the right-side image is detected it is edge detected image. Now, get back to detail filter. Detail filter enhance the details in the image. See here:but the filter name is dot filter of imagefilter dot detail.

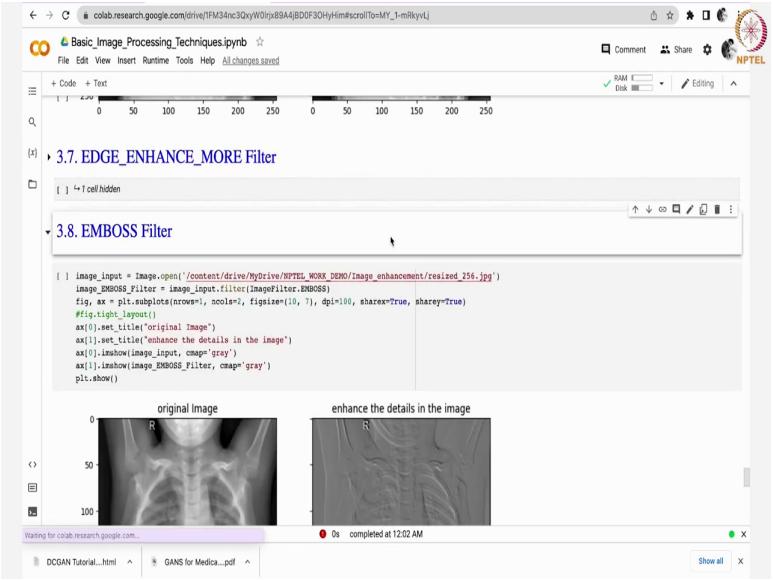
(Refer Slide Time: 21:09)

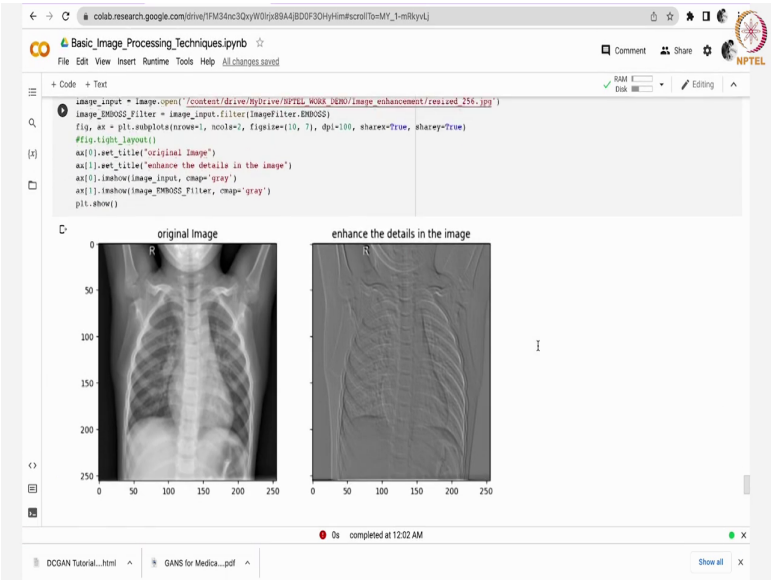


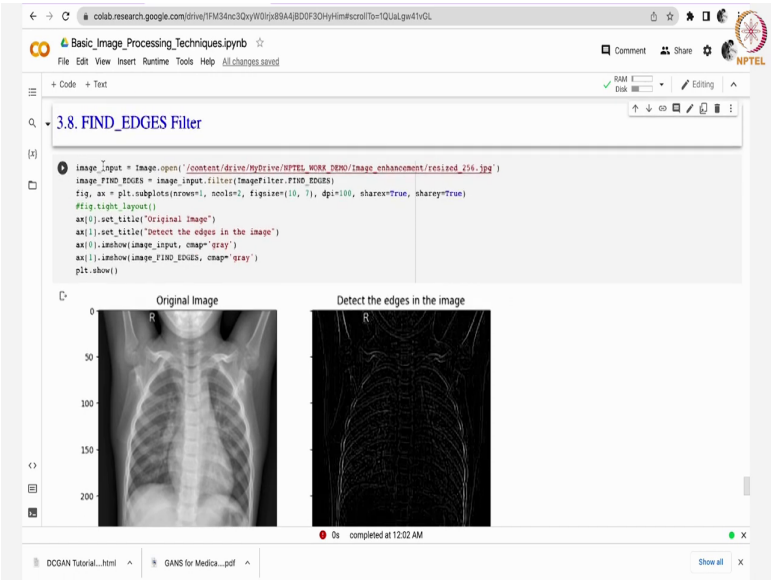


Likewise, we will apply based on our requirement "edge enhance" filter. See here, here we have detected the edges here it has clearly shown the edges enhance the details in the image. See using contour filter just we have detected the edges. But, in edge enhance filter you can see clearly edges are highlighted more and you can visualize well. That means the details of the edges have been enhanced.

(Refer Slide Time: 21:09)



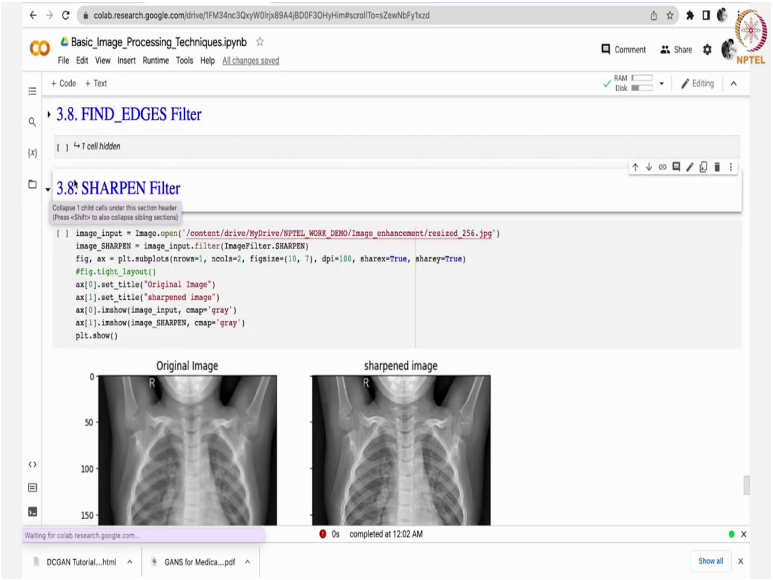


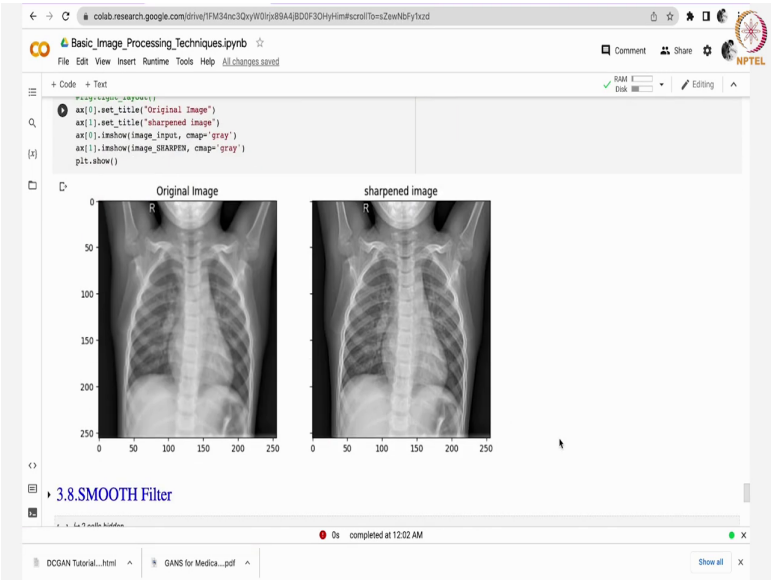


Similarly, you can go for edge enhance more filter. Next emboss filter. You just go through the code, just to change the load the image of your particular image just make some changes and get the output and see. Based on your requirements suppose if you want to use this filter and use this otherwise just ignore.

Find edges in a filter see here this is the original image detect the edges using find underscore edges filter. The summary is I have several filters of; there are several filters present in pillow library and you just have to use a particular filter based on your requirement and for that one particular code is already available you just have to load the path of the particular image correctly everything will happen later on.

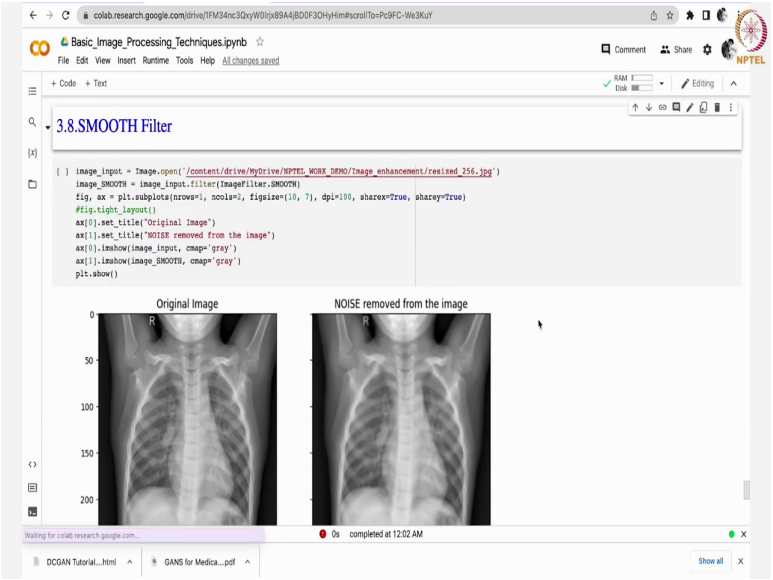
(Refer Slide Time: 22:30)

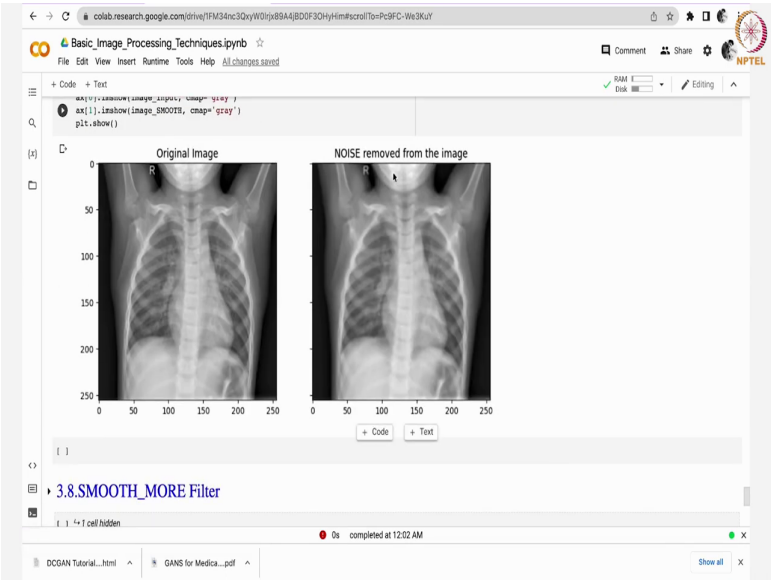


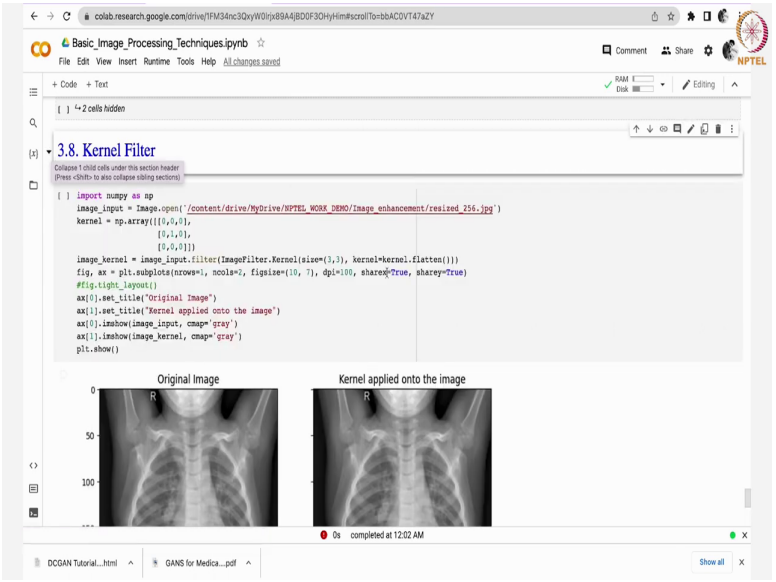


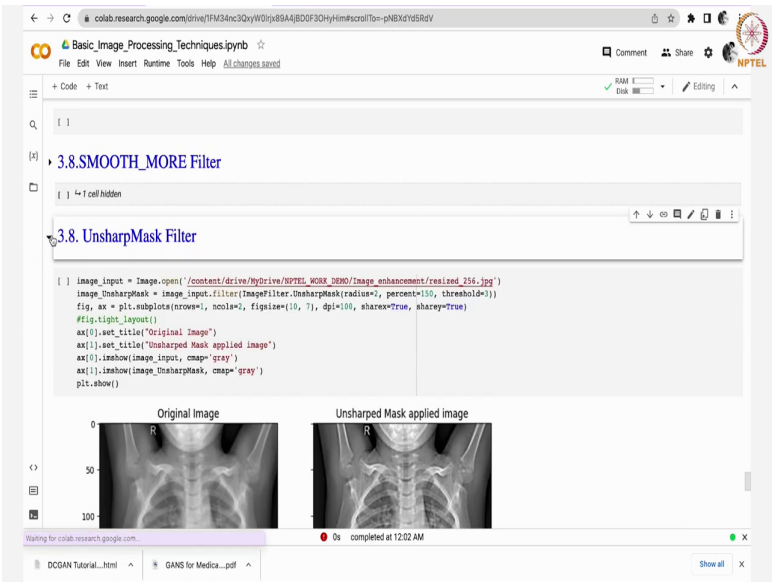
Now coming back to sharpen the details see. Sharpen filter also more or less equal to it enhancing the edges information only. In summary I can say that edges indicates the high frequency components in an image. That means we are enhancing the high frequency components.

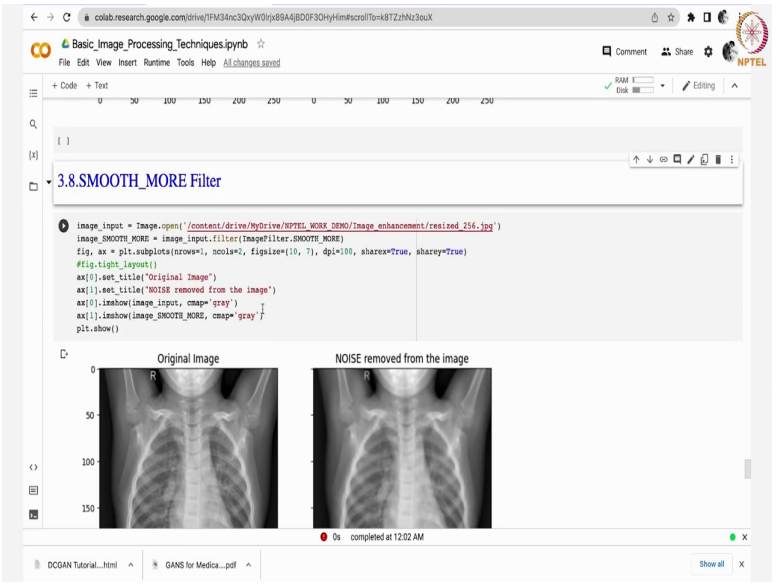
(Refer Slide Time: 22:51)

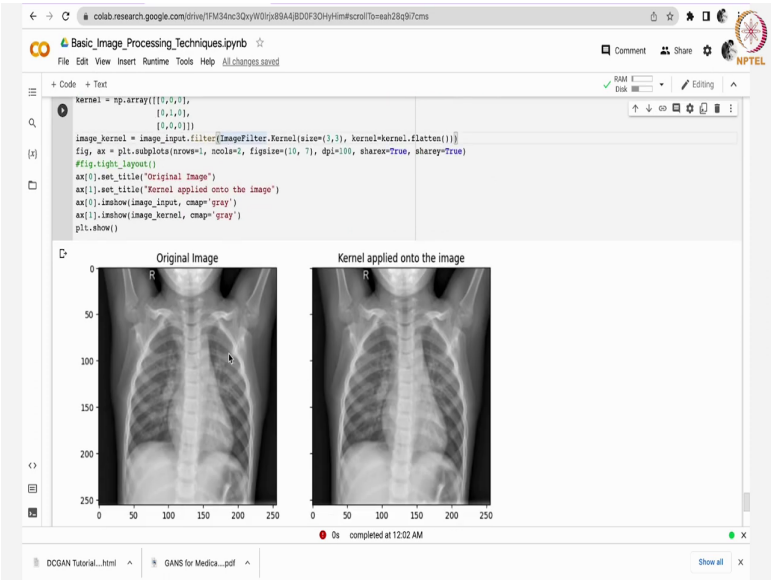





colab.research.google.com/drive/1FM34nc3QxyW0Irjx89A4jBD0F3OHyHim#scrollTo=bbAC0VT47aZY
Basic_Image_Processing_Techniques.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
NPTEL
+ Code + Text
RAM
Disk
Editing
[ ] ↳ 2 cells hidden
3.8. Kernel Filter
Collapse 1 child cells under this section header
(Press <Shift> to also collapse sibling sections)
[ ] import numpy as np
image_input = Image.open('/content/drive/MyDrive/NPTEL_WORK_DEMO/Image_enhancement/resized_256.jpg')
kernel = np.array([[0,0,0],
[0,1,0],
[0,0,0]])
image_kernel = image_input.filter(ImageFilter.Kernel(size=(3,3), kernel=kernel.flatten()))
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 7), dpi=100, sharex=True, sharey=True)
#fig.tight_layout()
ax[0].set_title("Original Image")
ax[1].set_title("Kernel applied onto the image")
ax[0].imshow(image_input, cmap='gray')
ax[1].imshow(image_kernel, cmap='gray')
plt.show()
Original Image
Kernel applied onto the image
0
50
100
0s completed at 12:02 AM
DCGAN Tutorial....html
GANS for Medica....pdf
Show all


colab.research.google.com/drive/1FM34nc3QxyW0Irjx89A4jBD0F3OHyHim#scrollTo=-pNBXdYd5RdV
Basic_Image_Processing_Techniques.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
NPTEL
+ Code + Text
RAM
Disk
Editing
[ ]
3.8.SMOOTH_MORE Filter
[ ] ↳ 1 cell hidden
3.8. UnsharpMask Filter
[ ] image_input = Image.open('/content/drive/MyDrive/NPTEL_WORK_DEMO/Image_enhancement/resized_256.jpg')
image_UnsharpMask = image_input.filter(ImageFilter.UnsharpMask(radius=2, percent=150, threshold=3))
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 7), dpi=100, sharex=True, sharey=True)
#fig.tight_layout()
ax[0].set_title("Original Image")
ax[1].set_title("Unsharped Mask applied image")
ax[0].imshow(image_input, cmap='gray')
ax[1].imshow(image_UnsharpMask, cmap='gray')
plt.show()
Original Image
Unsharped Mask applied image
0
50
100
Waiting for colab.research.google.com...
0s completed at 12:02 AM
DCGAN Tutorial....html
GANS for Medica....pdf
Show all

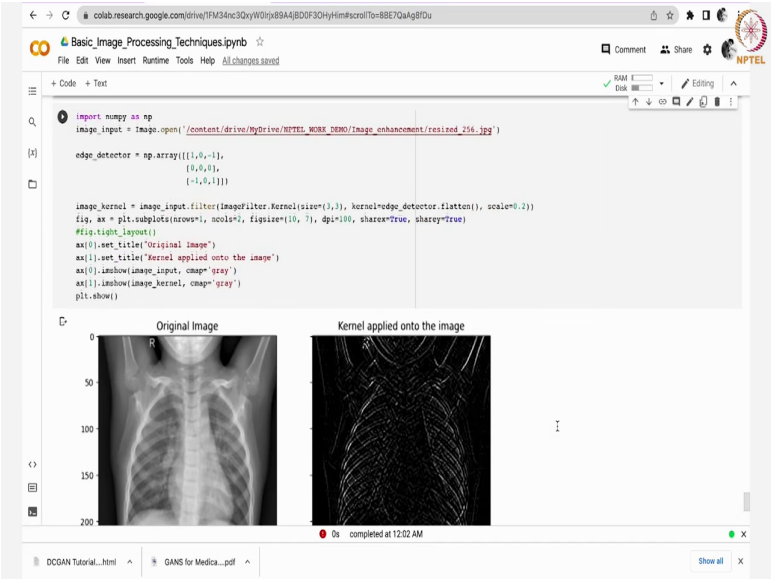

colab.research.google.com/drive/1FM34nc3QxyW0Irjx89A4jBD0F3OHyHim#scrollTo=k8TZzhNz3ouX
Basic_Image_Processing_Techniques.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
NPTEL
+ Code + Text
RAM
Disk
Editing
0 50 100 150 200 250 0 50 100 150 200 250
[ ]
3.8.SMOOTH_MORE Filter
image_input = Image.open('/content/drive/MyDrive/NPTEL_WORK_DEMO/Image_enhancement/resized_256.jpg')
image_SMOOTH_MORE = image_input.filter(ImageFilter.SMOOTH_MORE)
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 7), dpi=100, sharex=True, sharey=True)
#fig.tight_layout()
ax[0].set_title("Original Image")
ax[1].set_title("NOISE removed from the image")
ax[0].imshow(image_input, cmap='gray')
ax[1].imshow(image_SMOOTH_MORE, cmap='gray')
plt.show()
Original Image
NOISE removed from the image
0
50
100
150
0s completed at 12:02 AM
DCGAN Tutorial....html
GANS for Medica....pdf
Show all

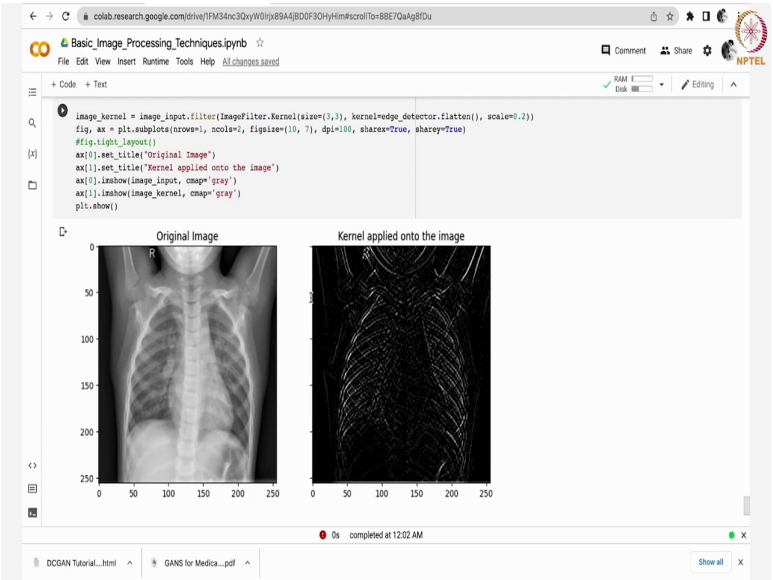


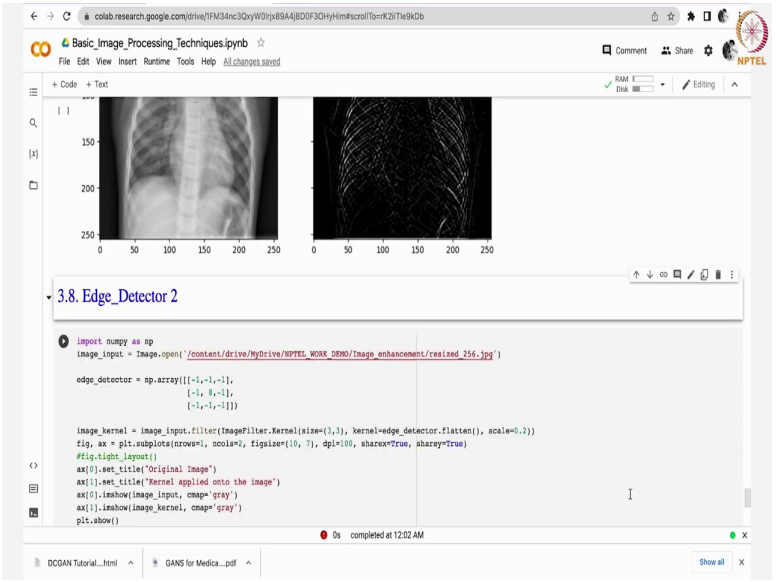
Next thing is smooth filter. Original image noise removed from the image next thing is smooth more filter, unsharp mask filter, kernel filter. Kernel filter is works in such a way that you can give the kernel value in these all the remaining things are like unsharp mask filter these are by default things unsharp masks imagefilter dot unsharp mask. Smooth filter see here image filter dot smooth more all are by default modules.

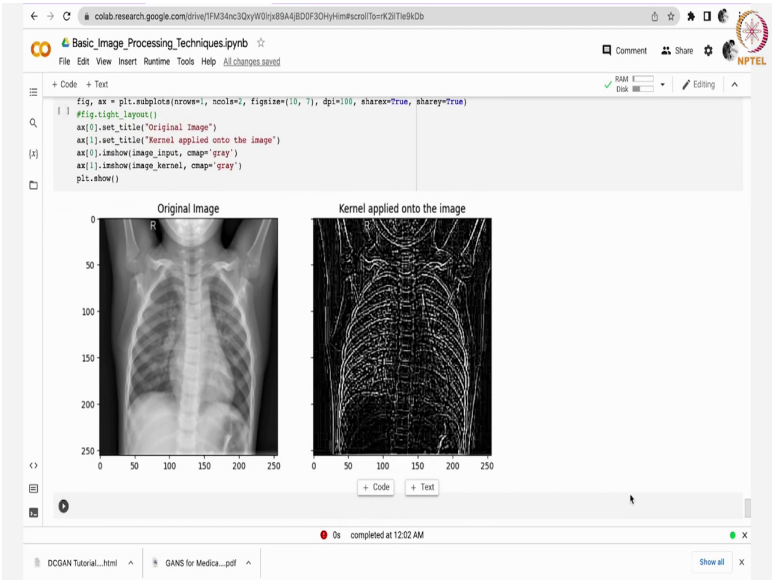
All these are all by default filters whereas if you want to use kernel filter you can design your own filter here and you can apply that particular kernel on this particular image and you can see the output. See here original image is on the left side on the right side kernel applied image.

(Refer Slide Time: 23:42)



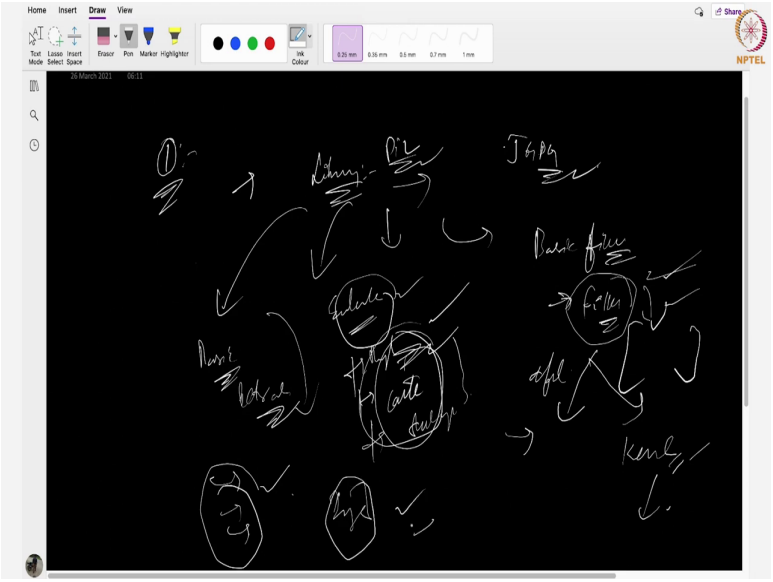






See when it comes to edge detector the applied filter is 1, 0, -1, 0, 0, 0, -1, 0, 1 this is the kernel thing that is up to 3 cross 3 size. See this is how it has detected the edges. Edge detector 2 this is how it has detected the edges likewise in summary I can say that.

(Refer Slide Time: 24:09)



In summary likewise in summary I can say that depending upon the requirement you just have to find out the particular library. Suppose, in our case we have used pillow library. Using the pillow library we have loaded the jpeg format related images. After loading that particular image we have done some basic modifications like resize operation, rotation operation, transpose operation, all that stuff.

Next thing is enhancement operations where we have focused on brightness increasing brightness increment similarly contrast increment and sharpness such operations. They work mostly on intensities and the last thing we discussed are basic filtering. Suppose, if you want to detect the edges on the in the image you just have to go through go through a particular filter and apply that filter on the image.

The code is already available you just have to go and give load your image and use that filter depending on your requirement. Next thing is even in filters also some default filters are there otherwise you can design your own kernel and apply that image this is the final summary. In short I can say that always basic image processing techniques are the fundamental step in any of the machine learning or deep learning problems especially when you deal with images. That is the final summary. Thanks a lot