# Bioimaging Assignment - 1


> Deep R Patel 😀
> B20CS087


---

## ↘️ Problem

- Feasibility of the DL model against the higher resolution i.e DL can give more approximation such that it is tend towards more resolving power microscopy generated outputs?
-

---

## ↘️ Given

We have the following functions:

- $g0.3(x, y)$
- $g0.7(x, y)$
- $f(x,y)$

Now:

$\varphi : g0.3(x, y) \rightarrow g0.7(x, y)$

We are having $g0.3(x,y)$ and $g0.7(x,y)$ but we are not having $f(x,y)$.

Beside this we also know that:

- g0.3(x, y): represent the observed images of a specimen f(x, y) viewed separately by
  the lense having 0.3 aperature.
- g0.7(x, y): represent the observed images of a specimen f(x, y) viewed separately by
  the lense having 0.7 aperature.

**Our task is to train a DL based model and learn the unknown mapping function φ**

---

## ↘️ Approach for DL based model

> I have thought of two models for this problem:
> 1. SRGAN model (Super Resolution Generative Adversarial Network)
> 2.CycleGAN model

**Super Resolution GAN (SRGAN)**

- Used for image super-resolution, which means increasing the resolution of an image while preserving its quality and details
- To deal with upscaling, SRGAN uses GAN architecture which consists of two main parts:
  - Generator: Takes low resolution image as input and generates a high-resolution version of it (generally by learning and trial & error estimation).It is trained to create images that are realistic enough to fool the discriminator.

    - Discriminator: To distinguish between the high resolution images generated by the generator ad the actual high-resolution images.
- Since, it is upscaling, thus some extra informatios is added which is non-linear (due to the nature of DL model) and it is generated randomly and learned from previous experince of the model

**CycleGAN**

- It is based on image-to-image translation task (which is more of what we have to do)
- It agains works by using generator and discriminator and they work together to learn how to transform low-resolution images to high-resolution ones. The generator network is trained to learn a mapping between low-resolution and high-resolution image pairs. However, unlike traditional GANs, CycleGAN does not require paired data, meaning that it can learn the mapping between low-resolution and high-resolution images without needing to know the exact relationship between them

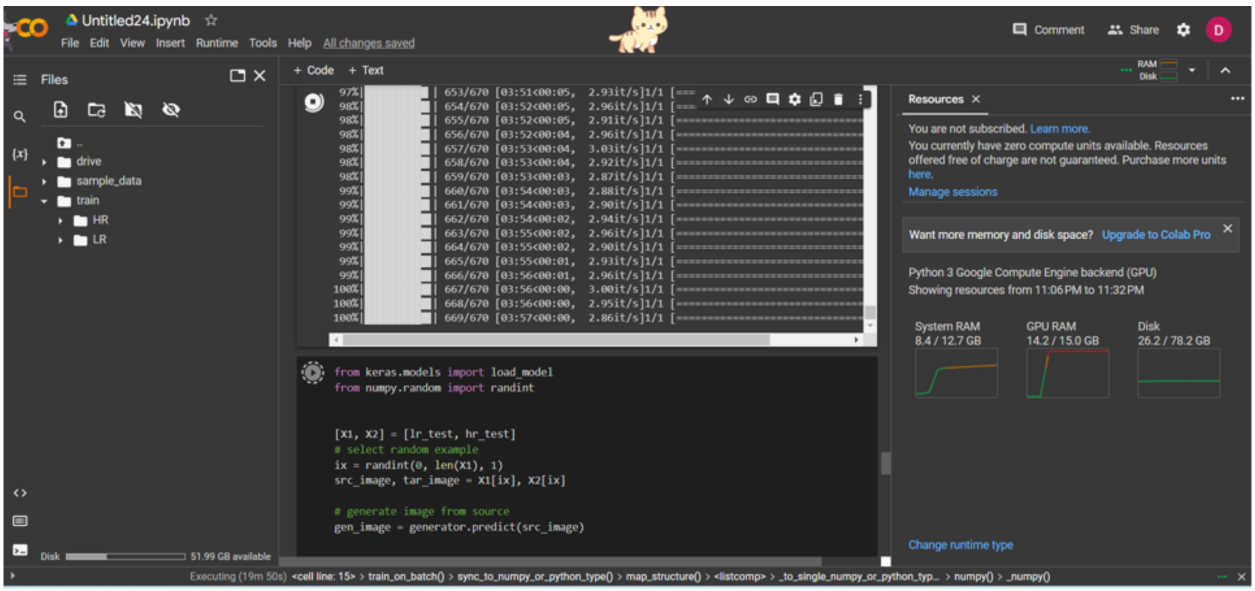
---

# ↘️ Observation

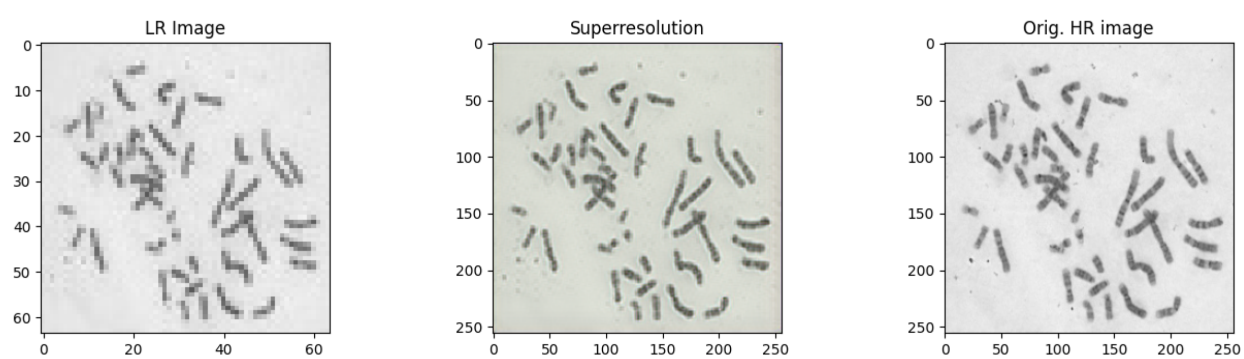
> g_loss : adversarial loss, which measures how well the generator can fool the discriminator. The adversarial loss encourages the generator to produce images that are similar to real images, so that the discriminator cannot tell them apart

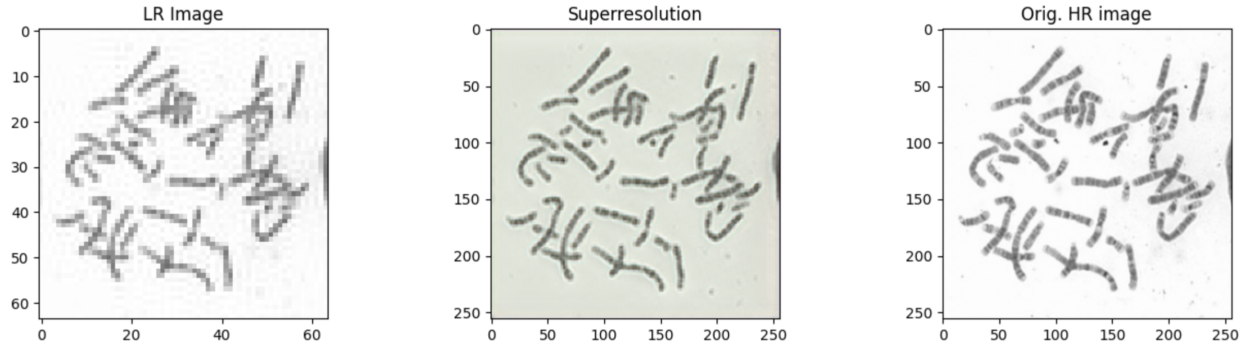
Basically , I have done the implementation of SRGAN (with 15GB GPU RAM and 12.7RAM, 1000 images were used and LR were sampled to 64x64 and HR were sampled to 256x256).

With that specs, even single epoch for CycleGAN was trained (since the HR and LR were both sampled to 256x256)



The images below are the result of testing of trained model (using 1000 random images)






Before this model, I have trained a new model on LR images (sampled at 32x32) and HR images (sampled at 128x128) and the g_error was around 34 after 5 epochs which is quite high and the results of that are stored in the corresponding jupyter notebook file which is attached along with the pdf

One thing which I observed is that, on incresing the epochs the loss is getting decreased (specifically talking of g_loss) , and if the LR image is sampled at high dimensions then the rate of decrement of loss is significantly higher.

And also I have attached the cycleGAN model's jupyter notebook (for ref)

Basically will return the resolved image which the user have provided the input for…

---

# ↘️ Conclusion

Since SRGAN is one of the top most model for the purpose of super resolution, but ultimately it approximates the missing high-frequency details even though it tries to apprximate by using the combination of content loss and adversarial loss (here content loss is the difference between the HR generated img and the original HR image in terms of pixel values). Even after so many epochs the loss can't tends to zero , thus SRGAN can be used to get the best approximate of the high resolved images of LR images but again it cant give exact HR image (after few epochs the loss becomes constant which means some high freq are missing or the missing frreuency which model predicted is not accurate).

While SRGAN is a top model for super resolution, it can only approximate missing high-frequency details using the combination of content loss and adversarial loss. Even after many epochs, the loss cannot reach zero, and the model cannot give an exact high-resolution image. Therefore, SRGAN can be used to get the best approximate of the high-resolved images of low-resolution images, but it cannot replace the need for higher resolution lenses.

Even we are not getting exact output, but sometimes it is better to use DL models as then expense of obtaining images are high and also,its time consumable.

> So,consider f(x,y)—>System—>g(x,y) , then if we try to obtain the f(x,y) then it’s not possible in both of the cases as we dont know what will be the frequency component and the DL model is actually learning  what can be the value (of 0/0 which is obtained after transforming the original sample piece to image and image to original one. Thus we can get what exactly the value was, but what we can do , is to estimate it by using several data

One more thing, I would like to mention, I haven’t used any pretrained models which were trained from other datasets , since this would generate bias , if we use the same model for training our dataset and eventually may lead to a bad estimation of LR —> HR. In order to decrease the loss, we can trained and save a model for particular sampling rate and then using this model for training other augmented datasets (augmentation of our dataset)