

## Search Google Colab

### Google Colab

<https://colab.research.google.com>

Web **Colab**, or "Colaboratory", allows you to write and execute Python in your browser, with . Zero configuration required; Access to GPUs free of charge; Easy sharing; Whether you're a ...



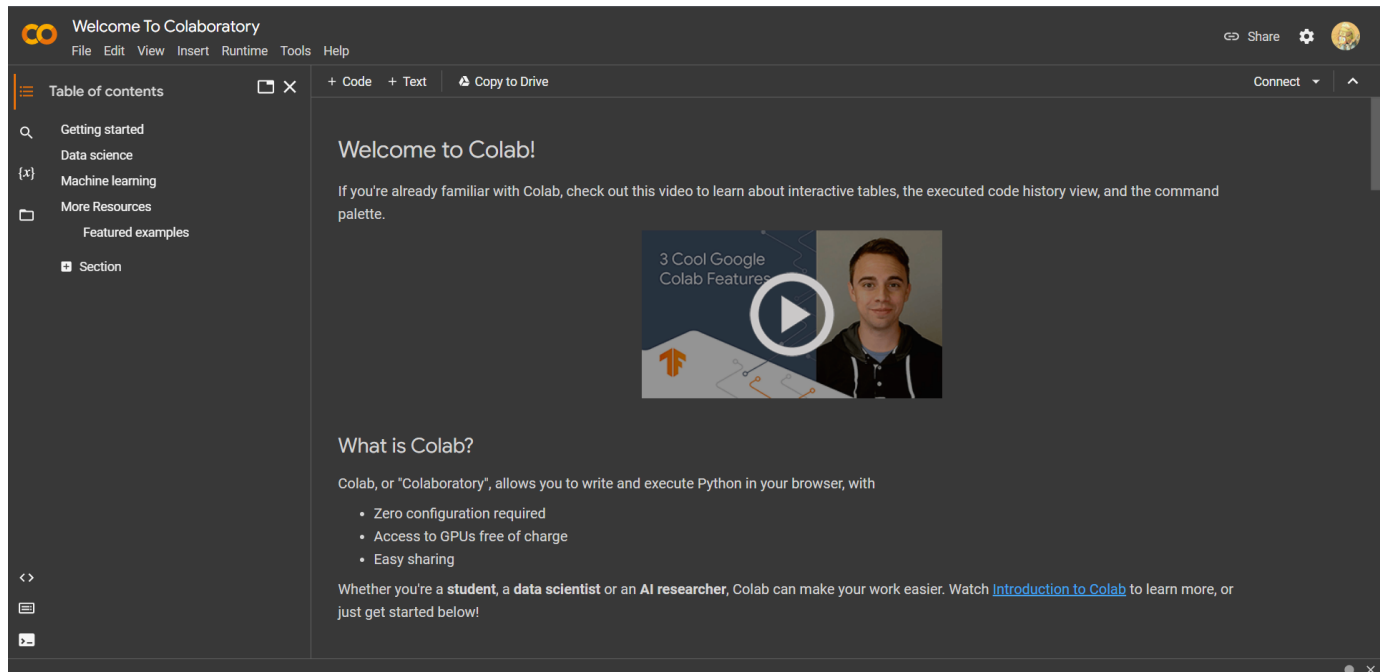
### Notebook

With Colab you can import an image dataset, train an image classifier on it, ...

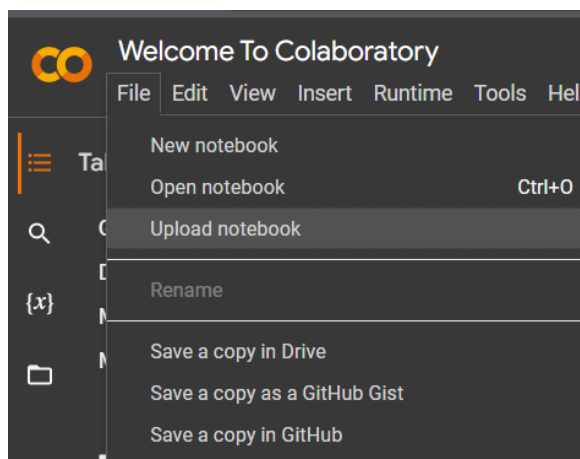
### Create a Gmail Account

Google Workspace starts at \$6 per user per month and includes the following: A ...

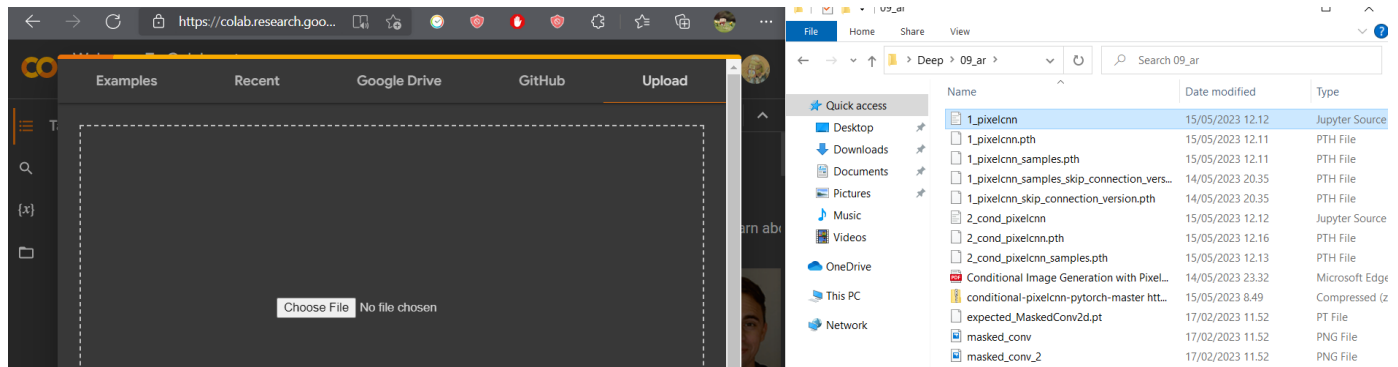
## You will see this page



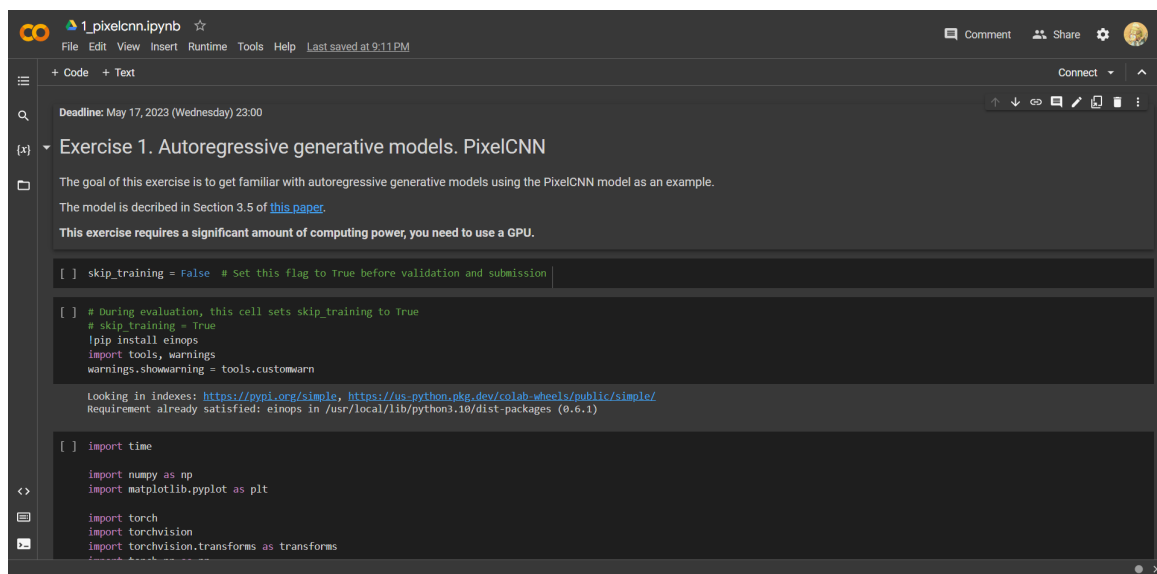
## Choose File > Upload notebook



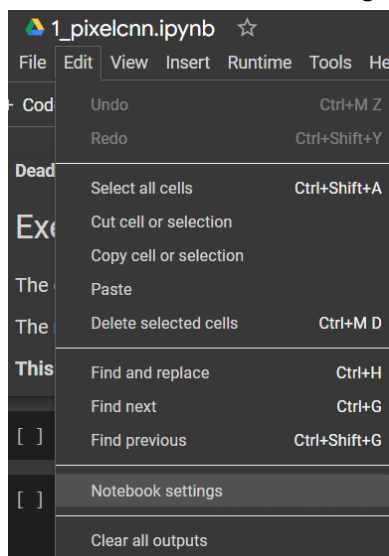
Drag the deep learning ipynb to the GPU, such as 1\_pixelcnn



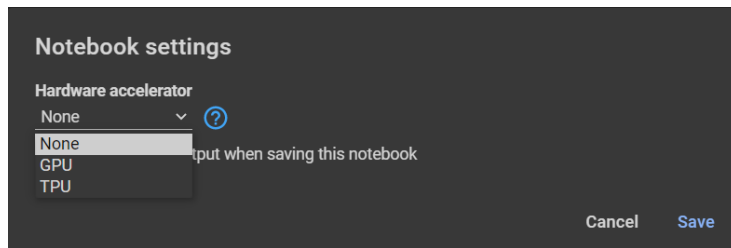
You will see this



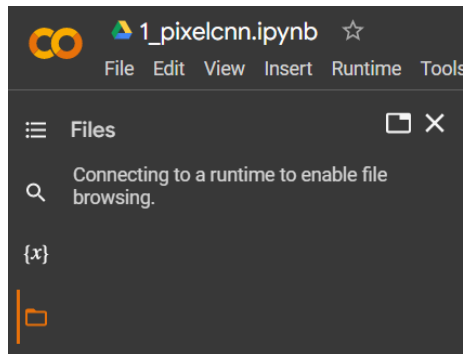
Click edit > Notebook settings



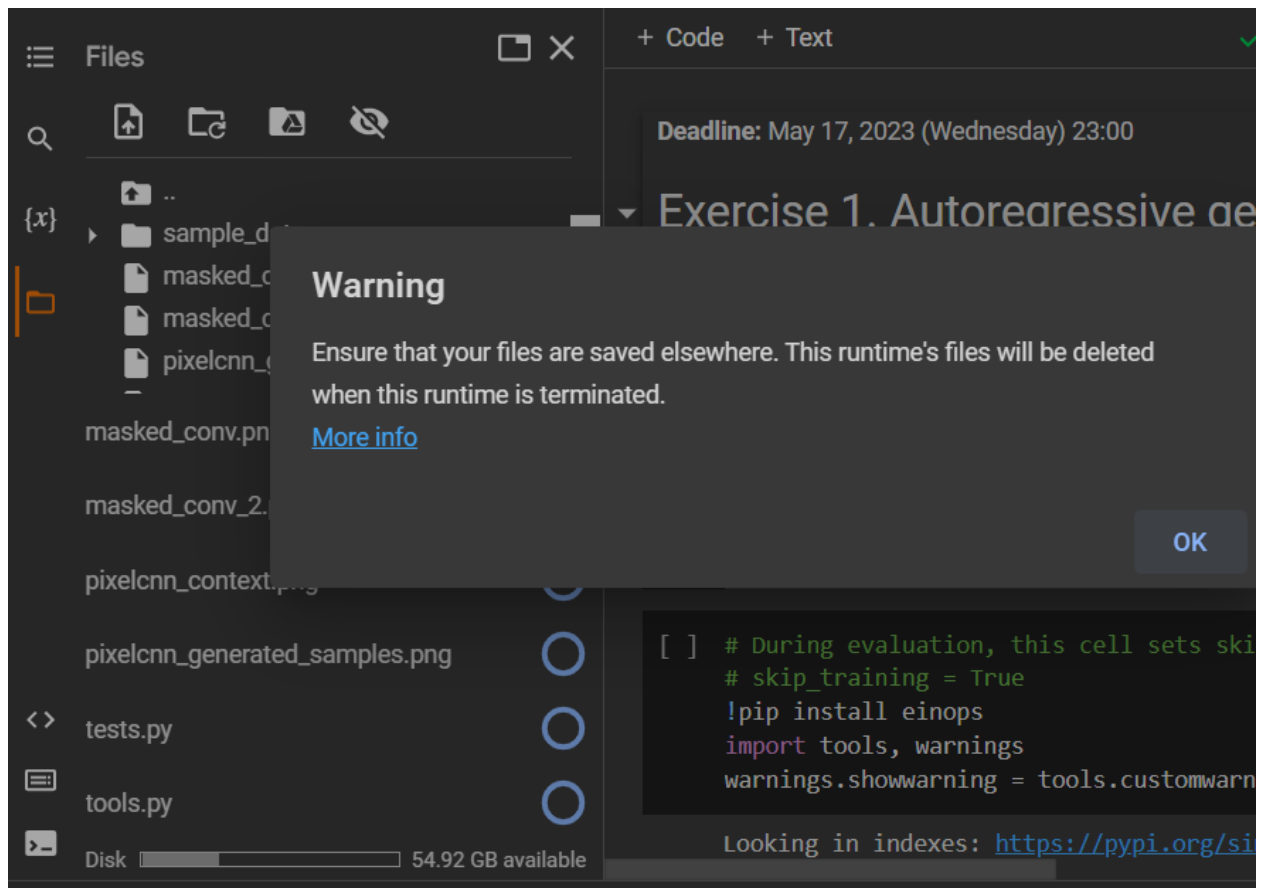
Choose GPU > Save



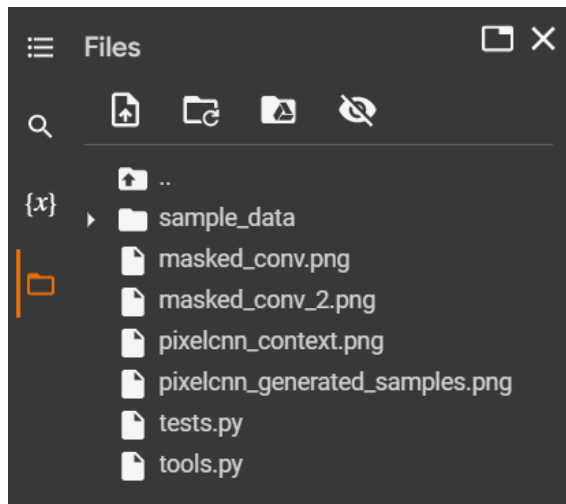
Click on the folder icon



Then drag all necessary files and dependencies from the DL assignment to this tab



You will have all dependencies uploaded as needed



Remember to download the ipynb file periodically in case you close the tab to save your progress, as when you close the tab, your work will be erased. However, clicking refresh is okay and safe as your data is still on the site.

```
# During evaluation, this cell sets skip_training to True
# skip_training = True
!pip install einops
import tools, warnings
warnings.showwarning = tools.customwarn
```

Then, add !pip install einops, since Google colab doesnt have this library.

Finally, choose device as GPU (cuda0)

```
# Select the device for training (use GPU if you have one)
#device = torch.device('cpu')
device = torch.device('cuda:0')
```

In init method of deep learning model, if you want to save a torch matrix data that is not part of nn modules, then you need to use register\_buffer like this

```
self.register_buffer('mask', mask)
```

Initialization of a torch matrix is set to the correct device (gpu or cpu) with device parameter

```
samples = torch.zeros((n_samples, images_channel, height, width), dtype=torch.float32, device=device)
```

You can also move an entire nn model to the device (gpu or cpu) with .to(device) method

```
# Create a model
net = PixelCNN(n_channels=64, kernel_size=7)
net.to(device)
```