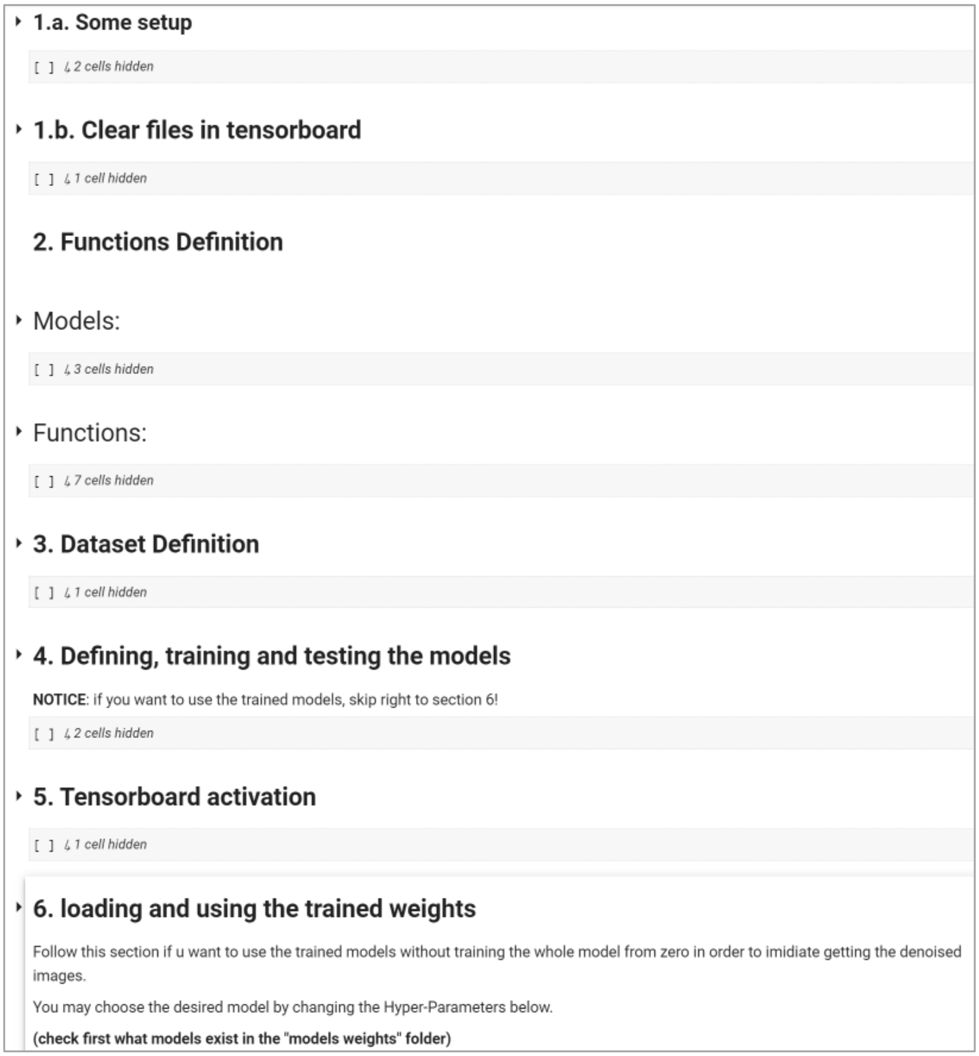
README

The code is divided into 6 main parts, as can be seen below:



1.a. Some setup

[ ] ↳ 2 cells hidden

1.b. Clear files in tensorboard

[ ] ↳ 1 cell hidden

2. Functions Definition

Models:

[ ] ↳ 3 cells hidden

Functions:

[ ] ↳ 7 cells hidden

3. Dataset Definition

[ ] ↳ 1 cell hidden

4. Defining, training and testing the models

NOTICE: if you want to use the trained models, skip right to section 6!

[ ] ↳ 2 cells hidden

5. Tensorboard activation

[ ] ↳ 1 cell hidden

6. loading and using the trained weights

Follow this section if u want to use the trained models without training the whole model from zero in order to imidiate getting the denoised images.

You may choose the desired model by changing the Hyper-Parameters below.

(check first what models exist in the "models weights" folder)

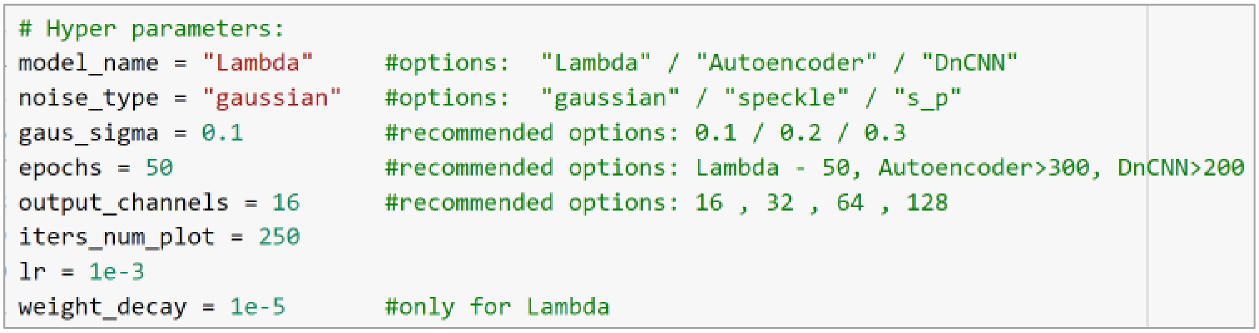
The code was run on “Google Colab”, the following instructions are related to this workspace. If desired, the code can be run on other platforms.

Before running the code, please follow the next steps:

1. Copy the “code” folders’ content to google drive.
2. Enter to the python file “final_project_deep_learning.ipynb” using google colab.
3. In part 1.a. notice to the 2 paths variable that are defined (saved_models_path and tensorboard_path). If it’s needed, change them in respect to the folder location in google drive.
4. In the toolbar enter to “Runtime”, “Change runtime type” and pick “GPU”.

There are 2 main options for using the code:

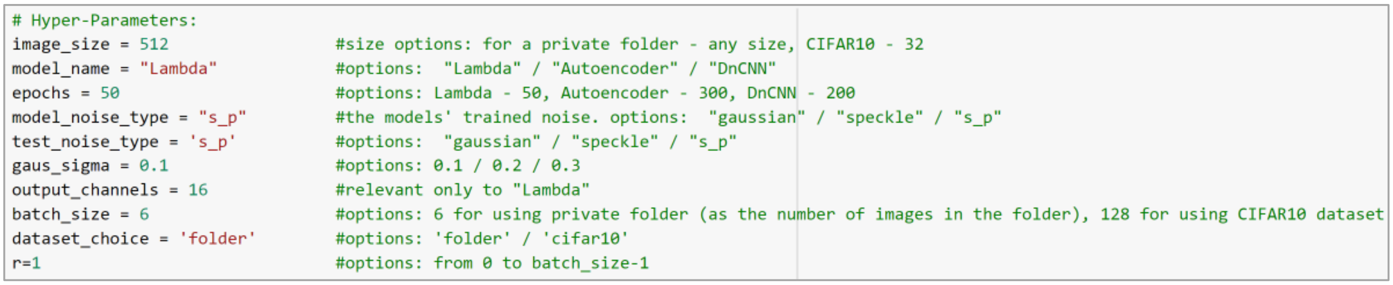
1. Training the desired model from scratch –
   - Go to part 4 and choose the desired model and hyper-parameters to train with. There are mentioned recommended options to choose from:



```
# Hyper parameters:
model_name = "Lambda"      #options:  "Lambda" / "Autoencoder" / "DnCNN"
noise_type = "gaussian"    #options:  "gaussian" / "speckle" / "s_p"
gaus_sigma = 0.1           #recommended options: 0.1 / 0.2 / 0.3
epochs = 50                #recommended options: Lambda - 50, Autoencoder>300, DnCNN>200
output_channels = 16       #recommended options: 16 , 32 , 64 , 128
iters_num_plot = 250
lr = 1e-3
weight_decay = 1e-5        #only for Lambda
```

   - Run parts 1-5

2. Loading and using the trained models' weights –
   - Go to part 6 and choose the desired model and the related hyper-parameters for loading the adjusted model weights.
     You can choose between testing a private dataset folder and CIFAR10 dataset.
     There are mentioned the options to choose from:



```
# Hyper-Parameters:
image_size = 512                 #size options: for a private folder - any size, CIFAR10 - 32
model_name = "Lambda"            #options:  "Lambda" / "Autoencoder" / "DnCNN"
epochs = 50                      #options: Lambda - 50, Autoencoder - 300, DnCNN - 200
model_noise_type = "s_p"         #the models' trained noise. options:  "gaussian" / "speckle" / "s_p"
test_noise_type = 's_p'          #options:  "gaussian" / "speckle" / "s_p"
gaus_sigma = 0.1                 #options: 0.1 / 0.2 / 0.3
output_channels = 16             #relevant only to "Lambda"
batch_size = 6                   #options: 6 for using private folder (as the number of images in the folder), 128 for using CIFAR10 dataset
dataset_choice = 'folder'        #options: 'folder' / 'cifar10'
r=1                              #options: from 0 to batch_size-1
```

   - Run only parts 1.a., 2 and 6
   - If you want to see the convergence graph of this model in tensorboard, first check in the folder "history"/"tensorboard history" If it exists. If so, copy the file to the folder "tensorboard" (make sure that this is the only file that exist there) and run also part 5