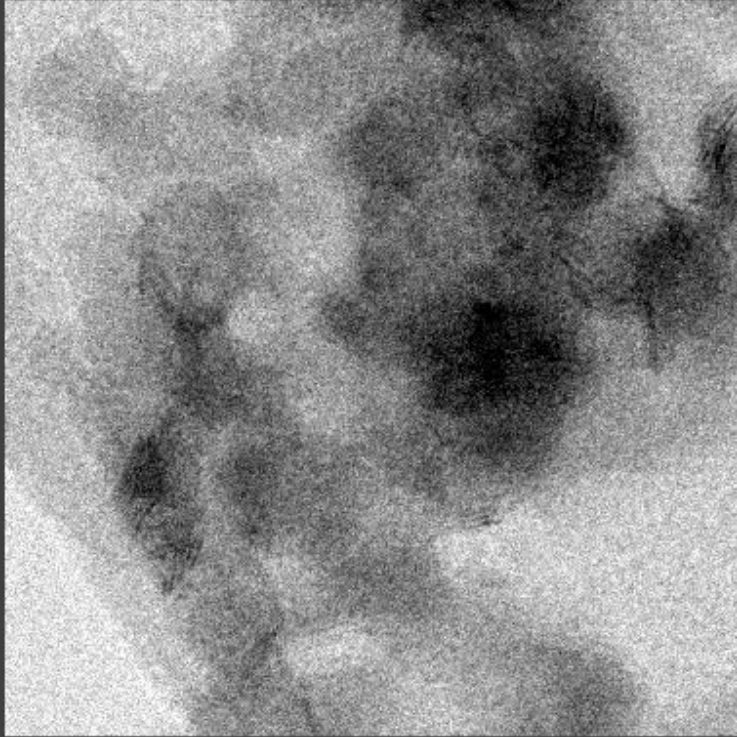
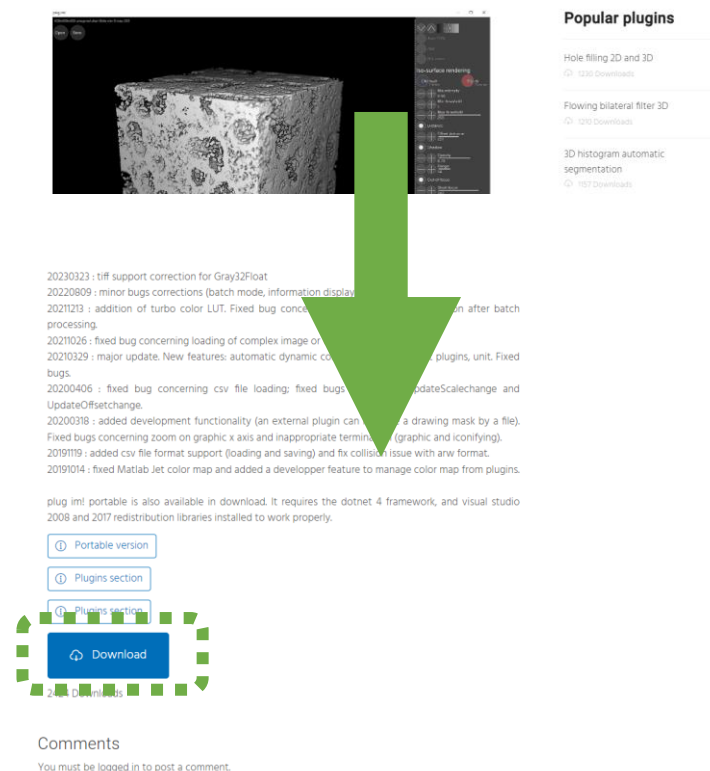
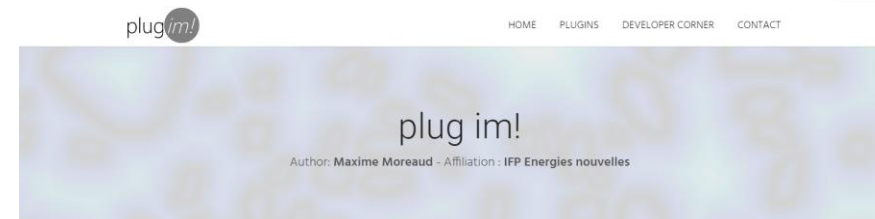
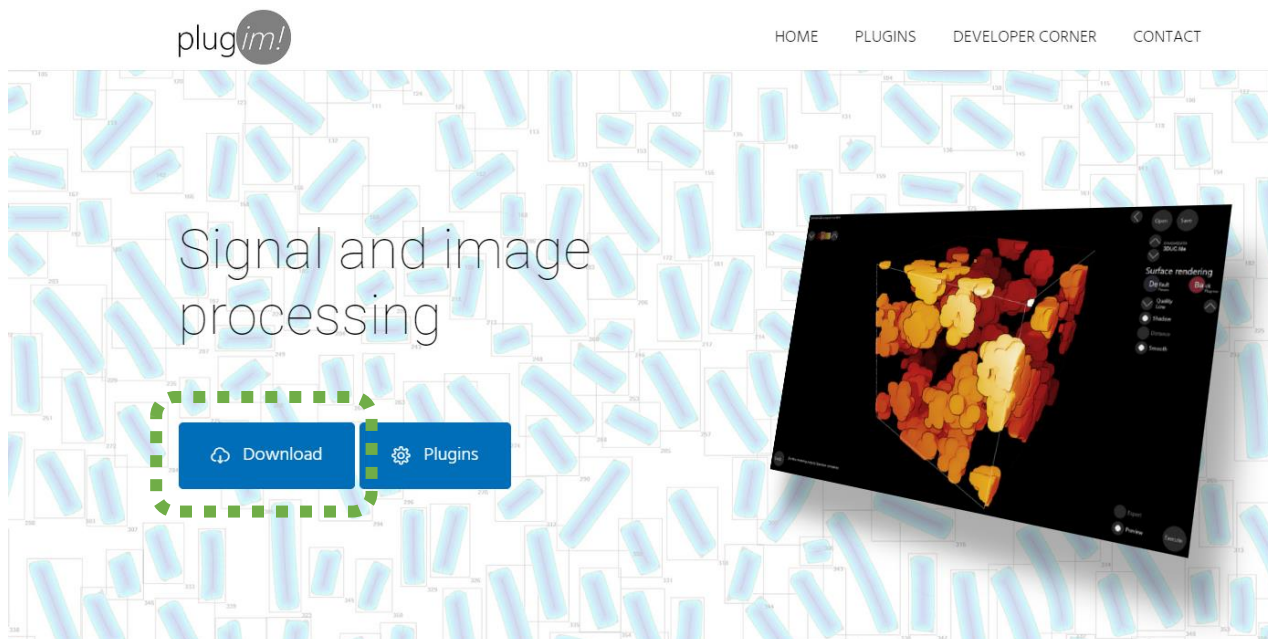


Noise reduction with deep learning, a comprehensive tutorial with plug im!



Download and install plug im software

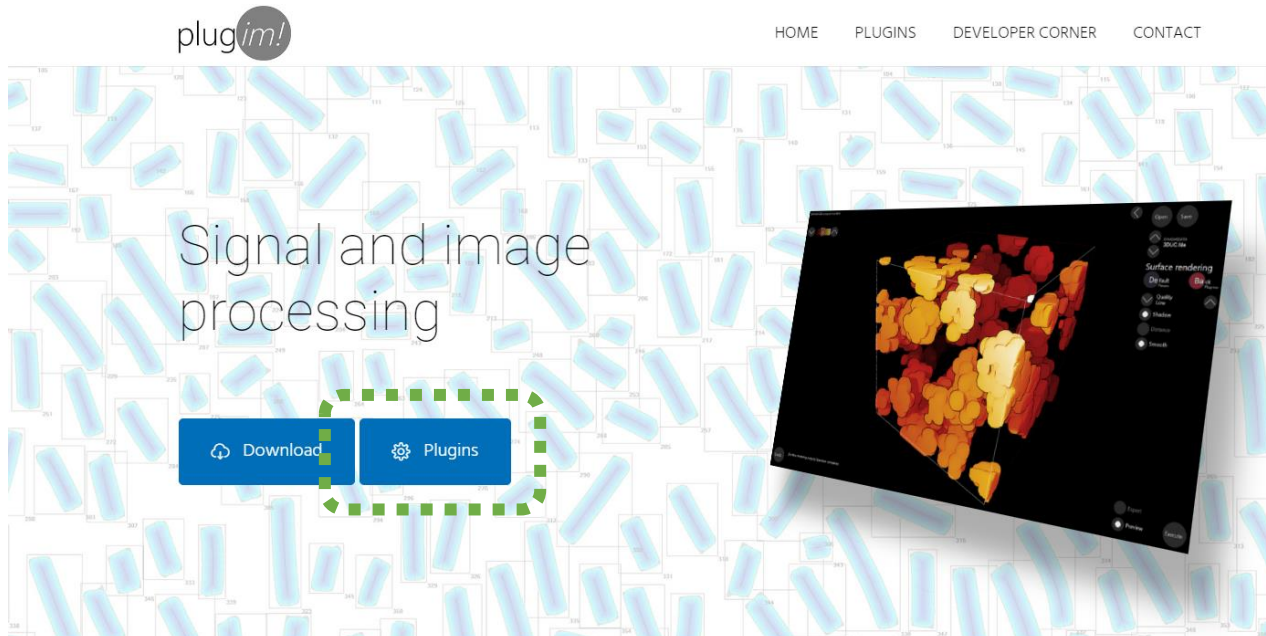
 www.plugim.fr



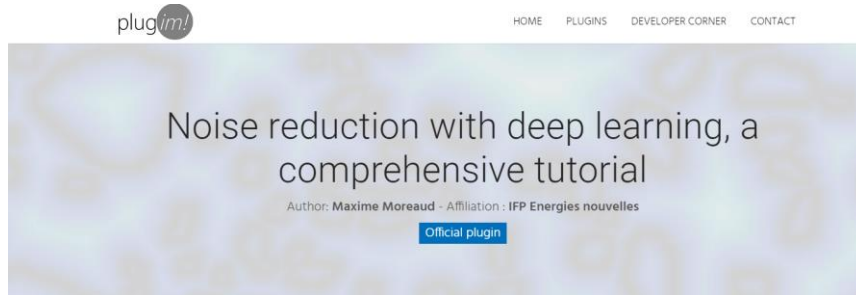
 Follow installation procedure

Download "Noise reduction" Package (1/2)

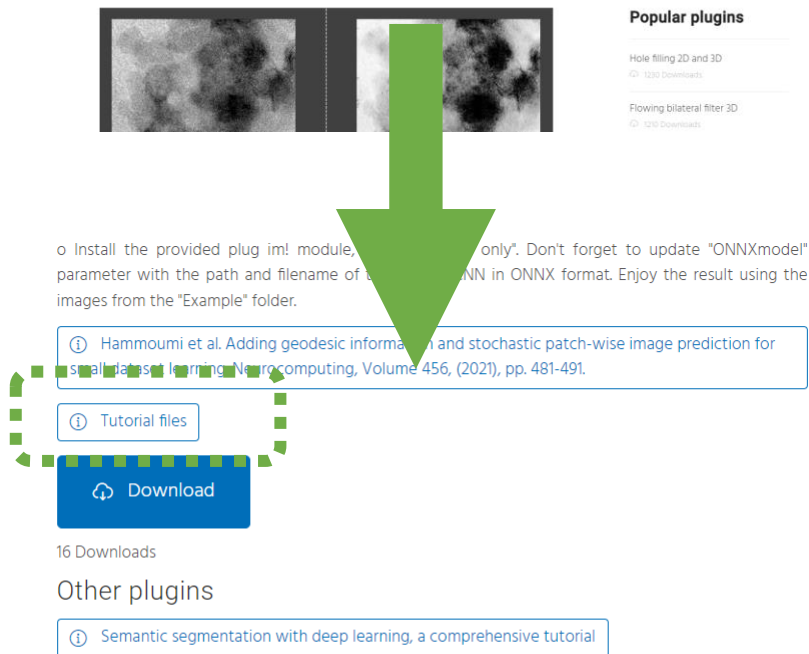
www.plugim.fr



Download "Noise reduction" Package (2/2)



● Package :



- 📁 _Examples
- 📁 _InstallPythonONNXconversion
- 📁 _InstallPythonTraining
- 📁 ONNXconversion
- 📁 Training

Comments

You must be [logged in](#) to post a comment.

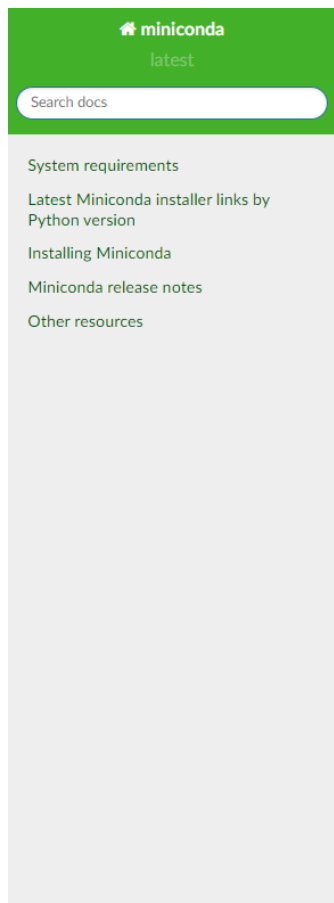
<https://www.plugim.fr/plugin/119>

Tutorial steps

- Training
 - Miniconda installation
 - Python environnement installation
 - Jupyter Notebook
- ONNX conversion
 - Python environnement installation
 - Jupyter Notebook
- Inference with plug im!
 - Plugin installation

Training / Miniconda installation

- Install Miniconda3 for Windows 64-bit with default instructions
- <https://docs.conda.io/projects/miniconda/en/latest/index.html>



🏠 / Miniconda

Miniconda

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only conda, Python, the packages they both depend on, and a small number of other useful packages (like pip, zlib, and a few others). If you need more packages, use the `conda install` command to install from thousands of packages available by default in Anaconda's public repo, or from other channels, like conda-forge or bioconda.

Is Miniconda the right conda install for you? The [Anaconda](#) or [Miniconda](#) page lists some reasons why you might want one installation over the other.

- [System requirements](#)
- [Latest Miniconda installer links by Python version](#)
- [Installing Miniconda](#)
- [Miniconda release notes](#)
- [Other resources](#)

Latest Miniconda installer links

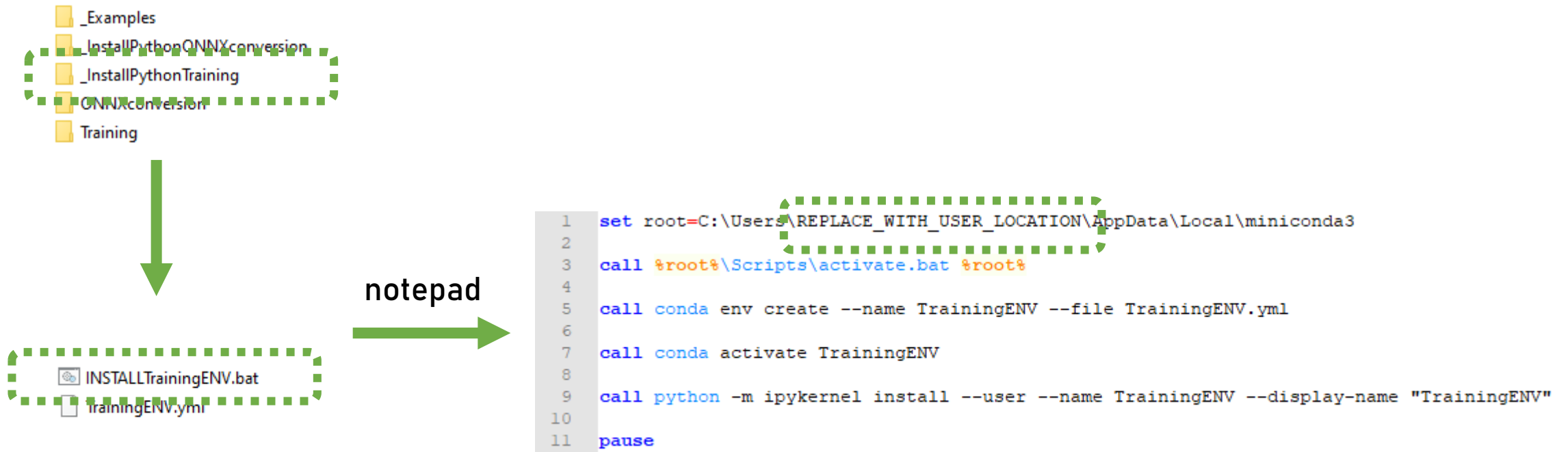
This list of installers is for the latest release of Python: 3.11.4. For installers for older versions of Python, see [Other installer links](#). For an archive of Miniconda versions, see <https://repo.anaconda.com/miniconda/>.

Latest - Conda 23.5.2 Python 3.11.4 released July 13, 2023

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	00e8370542836862d4c790aa8966f1d7344a8add4b766004febcb23f40e2914
macOS	Miniconda3 macOS Intel x86_64-bit bash	1622e7a0fa60a7d3d892c2d8153b54cd6ffe3e6b979d931320ba56bd52581d4b
	Miniconda3 macOS Intel x86_64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c031d41791b44612bbb6a7a1a6b4
	Miniconda3 macOS Apple M1 64-bit bash	c8f436dbde130f171d39dd7b4fca669c223f130ba7789b83959adc1611a35644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f3b6e8ae2b65bdfc8370a6be812b564ff9f40bdc4e0b027f84bf9h4fe5

Training / Python environment installation

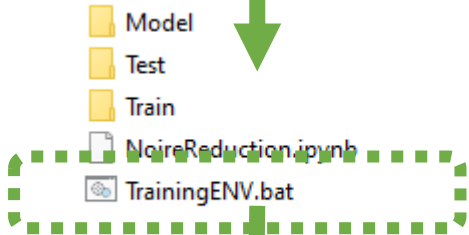
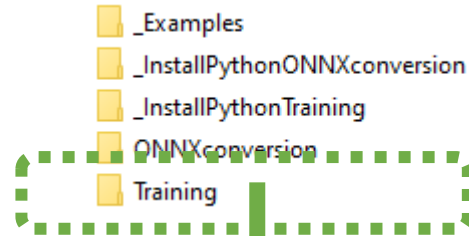
● Update _InstallPythonTraining / INSTALLTrainingENV.bat



● Double click on INSTALLTrainingENV.bat

Training / Jupyter notebook (1/2)

● Update Training / TrainingENV.bat



notepad

```
1 set root=C:\Users\REPLACE_WITH_USER_LOCATION\AppData\Local\Continuum\anaconda3
2
3 call %root%\Scripts\activate.bat %root%
4
5 call conda activate TrainingENV
6
7 call jupyter notebook
8
9 pause
```

● Double click on TrainingENV.bat



Files

Running

Clusters

Select items to perform actions on them.

<input type="checkbox"/>	0	<input type="button" value="v"/>	<input type="button" value="f"/>	/
<input type="checkbox"/>	<input type="button" value="f"/>	Model		
<input type="checkbox"/>	<input type="button" value="f"/>	Test		
<input type="checkbox"/>	<input type="button" value="f"/>	Train		
<input type="checkbox"/>	<input type="button" value="f"/>	NoireReduction.ipynb		
<input type="checkbox"/>	<input type="button" value="f"/>	TrainingENV.bat		

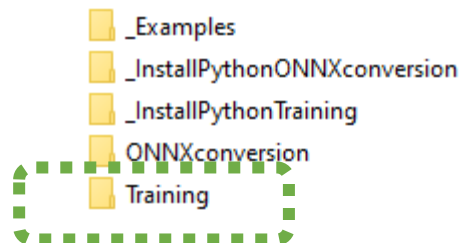
Training / Jupyter notebook (2/2)

Update Notebook

```

Entrée [ ]: import os
import sys
from tqdm import tqdm,tqdm_notebook
import tensorflow as tf
from keras.models import Model, load_model
from keras.layers import Input, BatchNormalization, Activation, Dropout
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from keras.optimizers import Adam
from skimage.transform import resize
import keras
from keras.preprocessing.image import img_to_array, load_img
from keras import backend as K
import math
import numpy as np
import matplotlib.pyplot as plt
import PIL
import warnings
import time
from PIL import Image
import matplotlib
import imageio
import keras2onnx
warnings.filterwarnings('ignore')

from keras.preprocessing.image import ImageDataGenerator
from matplotlib.pyplot import imread, imshow, subplots, show
from keras.preprocessing.image import save_img
from distutils.version import LooseVersion
    
```



GPU configuration check

File parameters

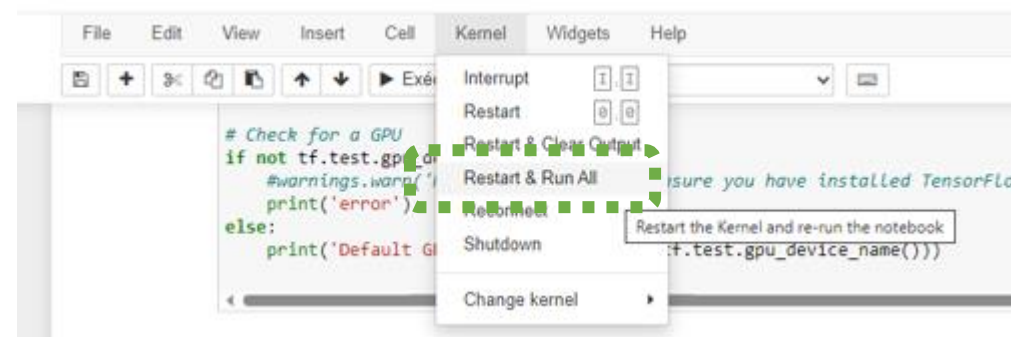
```

Entrée [ ]: vardir=r"C:\REPLACE_WITH_CORRECT_DESTINATION/
Train_images = os.path.join(vardir+"Train\image/")
Train_mask = os.path.join(vardir+"Train\label/")
Val_images = os.path.join(vardir+"Test\image/")
Val_mask = os.path.join(vardir+"Test\label/")
Model_path = os.path.join(vardir+"Model/")
Model_name = r"Denoise.h5"

#Names of all images in training and validation paths
ids = next(os.walk(Train_images))[2]
ids_validation = next(os.walk(Val_images))[2]

#parameters
patch_size = 128
pas = 64
    
```

Execute Notebook

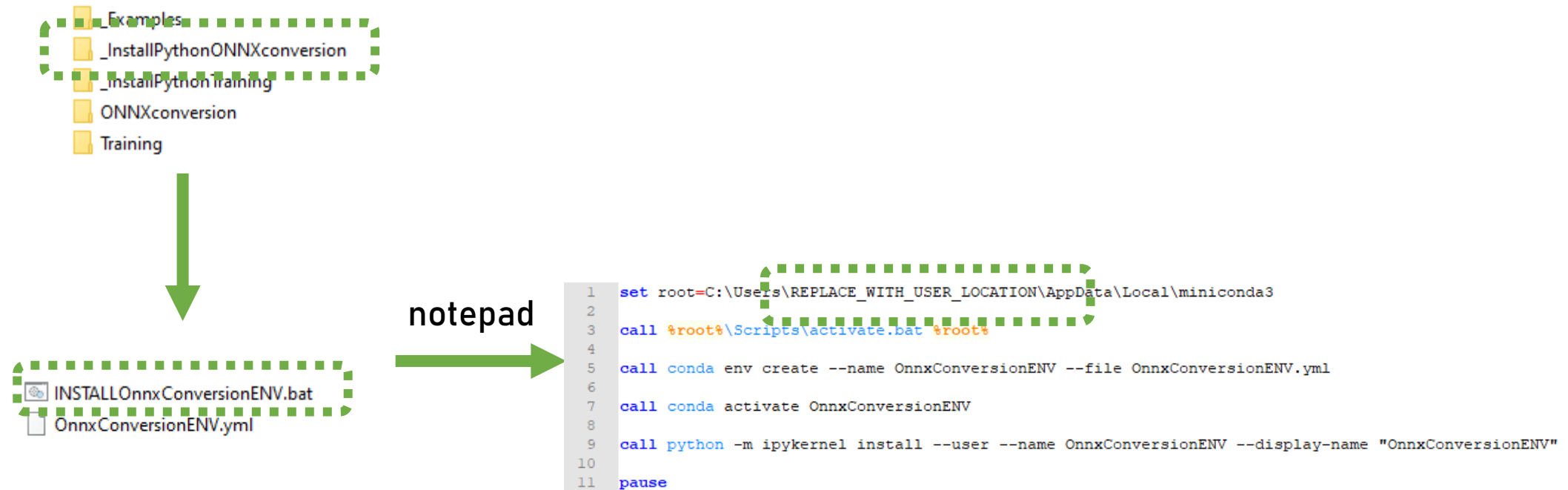


Tutorial steps

- Training
 - Miniconda installation
 - Python environnement installation
 - Jupyter Notebook
- ONNX conversion
 - Python environnement installation
 - Jupyter Notebook
- Inference with plug im!
 - Plugin installation

ONNX conversion / Python environment installation

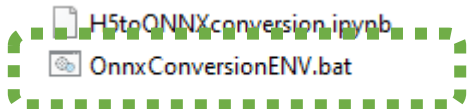
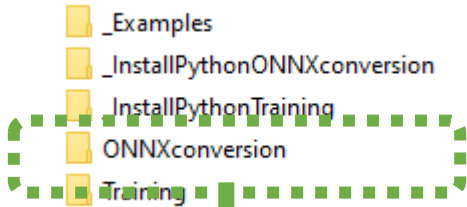
● Update _InstallPythonONNXconversion / _INSTALLOnnxConversionENV.bat



● Double click on INSTALLOnnxConversionENV.bat

ONNX conversion / Jupyter notebook (1/2)

● Update ONNXconversion / OnnxConversion.bat



notepad

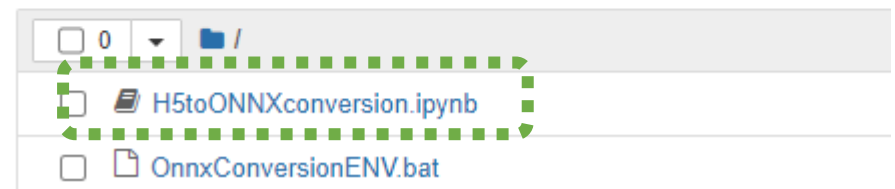
```
1 set root=C:\Users\REPLACE_WITH_USER_LOCATION\AppData\Local\miniconda3
2
3 call %root%\Scripts\activate.bat %root%
4
5 call conda activate OnnxConversionENV
6
7 call jupyter notebook
8
9 pause
```

● Double click on OnnxConversionENV.bat



Files Running Clusters

Select items to perform actions on them.



ONNX conversion / Jupyter notebook (2/2)

● Update Notebook



jupyter H5toONNXconversion (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help

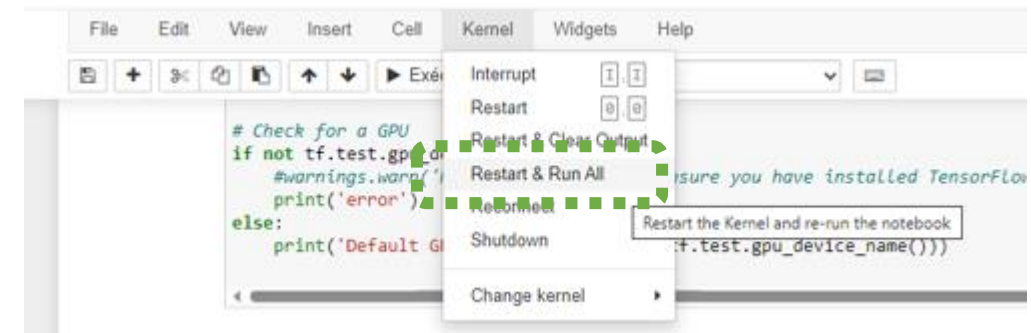
Exécuter Code

```
Entrée [ ]: import os
os.environ['TF_KERAS'] = '1'
import onnx
import tensorflow as tf
import keras2onnx

model = tf.keras.models.load_model("D:\\Regp128Unet.h5", compile=False)

onnx_model = keras2onnx.convert_keras(model, model.name)
file = open("D:\\Regp128Unet.onnx", "wb")
file.write(onnx_model.SerializeToString())
file.close()
```

● Execute Notebook

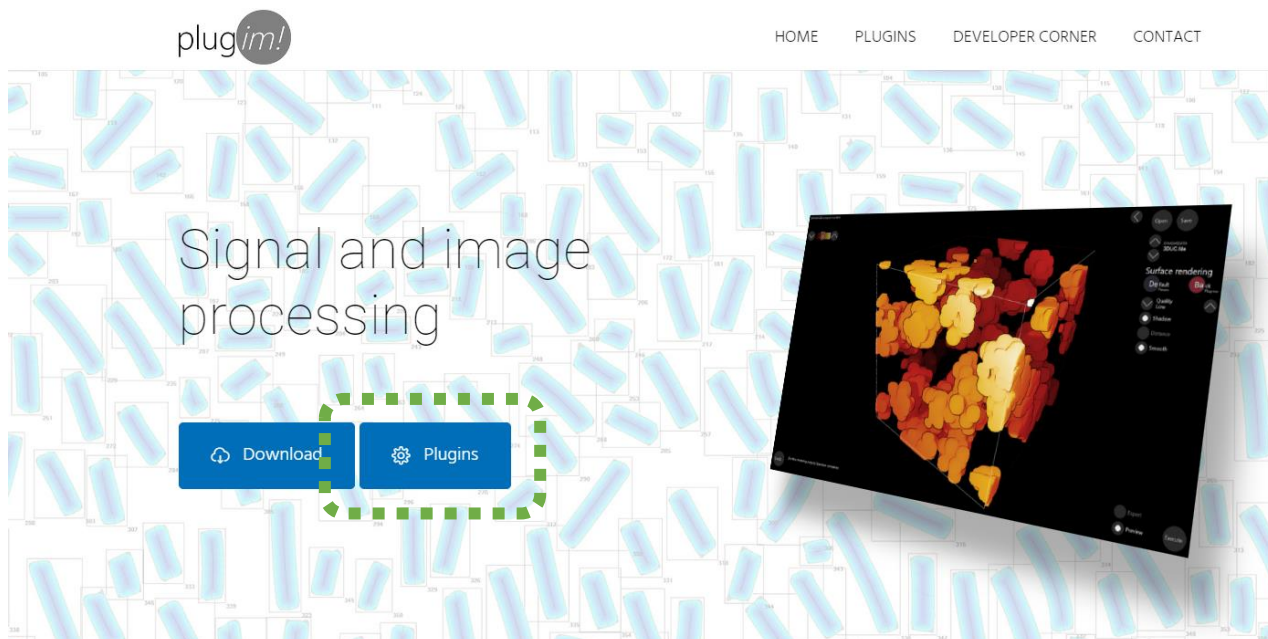


Tutorial steps

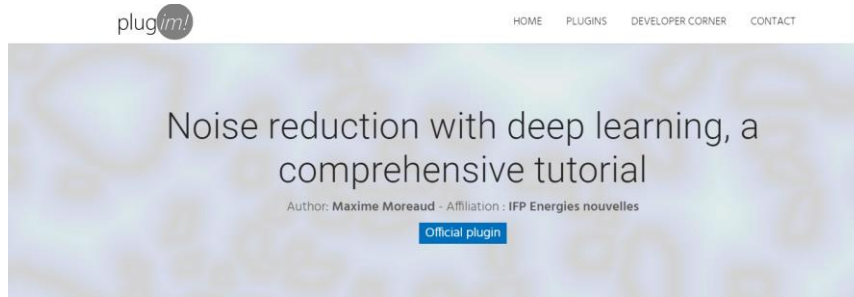
- Training
 - Miniconda installation
 - Python environnement installation
 - Jupyter Notebook
- ONNX conversion
 - Python environnement installation
 - Jupyter Notebook
- Inference with plug im!
 - Plugin installation

Download "Noise reduction" plugin

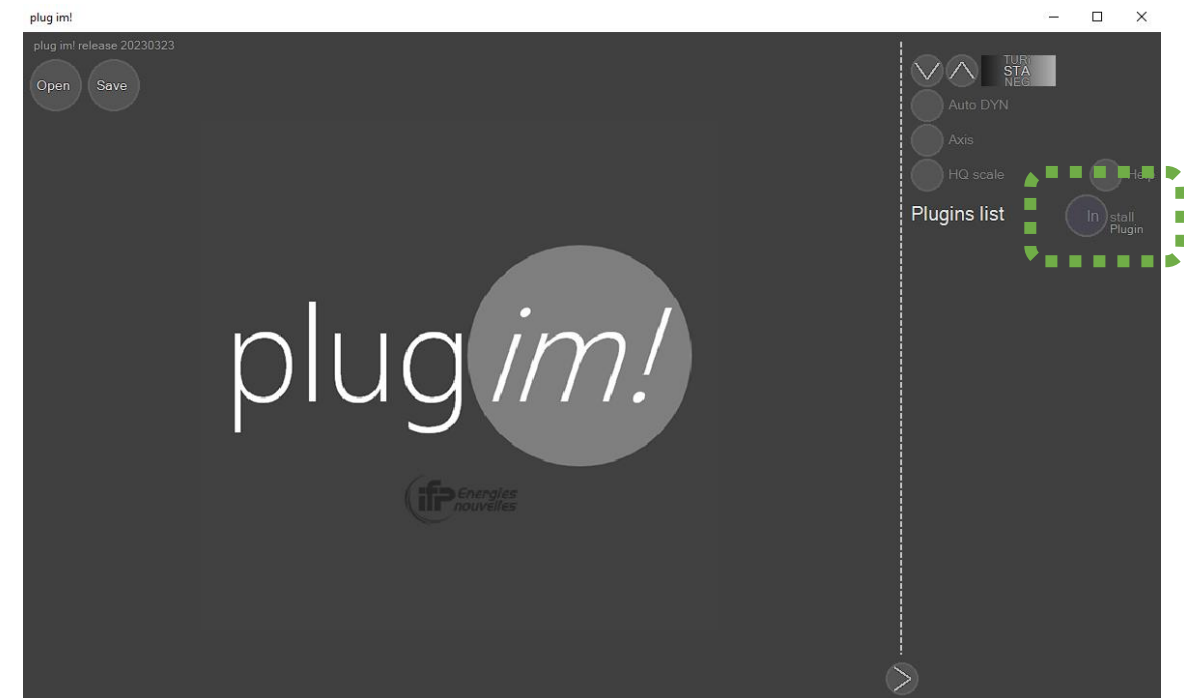
www.plugim.fr



Download "Noise reduction" plugin



● Install downloaded zip file inside plug im!



<https://www.pluginim.fr/plugin/119>

Try Noise reduction

- Load an image from _Examples
- Update ONNX model file parameter
- Enjoy!

