ACCELERER LE TEMPS DE TRAITEMENT SOUS PIXINSIGHT EN UTILISANT LA PUISSANCE DE SA CARTE GRAPHIQUE NVIDIA

Réalisé par Steph@stro
https://stephastro.go.zd.fr/

Mars 2023

Sommaire

1 INTRODUCTION 2

2 RESSOURCES 2

3 METTRE A JOUR SA VERSION DU DRIVER NVIDIA 3

4 TEMPS DE TRAITEMENT AVANT MODIFICATION 3

5 LOGICIELS A TELECHARGER 4

5.1 CUDA Toolkit 11.8 4
5.2 Cudnn 5
5.3 Tensorflow 7
5.4 Zlib DLL 7

6 INSTALLATION ET CONFIGURATION 7

6.1 Étape 1 : Remplacer le tensorflow.dll 7
6.2 Étape 2 : Installer CUDA 7
6.3 Étape 3 : Copier/coller fichiers cuDNN 10
6.4 Étape 4 : Copier/coller fichier Zlib 10

7 MODIFIER LES VARIABLES D'ENVIRONNEMENT 11

8 RESULTAT 13

1 INTRODUCTION

Ce tutorial a pour but d'accélérer le temps de traitement sous Pixinsight en utilisant la puissance de sa carte graphique au lieu de celle du processeur.

Cela ne fonctionne qu'avec les cartes graphiques équipées en Nvidia.

2 RESSOURCES

Ressources qui m'ont aidé pour la rédaction de ce tuto :

https://darkskies.space/pixinsight-starnet-cuda/

Traduction google : https://darkskies-space.translate.goog/pixinsight-starnet-cuda/?_x_tr_sl=auto&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=wapp

https://www.williamliphotos.com/starnet-cuda

Traduction google : https://www-williamliphotos-com.translate.goog/starnet-cuda?_x_tr_sl=auto&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=wapp&_x_tr_hist=true

https://rikutalvio.blogspot.com/2023/02/pixinsight-cuda.html

Traduction google : https://rikutalvio-blogspot-com.translate.goog/2023/02/pixinsight-cuda.html?_x_tr_sl=auto&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=wapp

https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html#install-zlib-windows

Traduction google : https://docs-nvidia-com.translate.goog/deeplearning/cudnn/install-guide/index.html?_x_tr_sl=auto&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=wapp#install-zlib-windows

http://www.astrosurf.com/topic/159388-cuda-et-pixinsight/

https://avex-asso.org/ips/topic/8190-starnet-avec-son-gpu-au-lieu-du-cpu-pour-pix-ou-sans-pix/

3 METTRE A JOUR SA VERSION DU DRIVER NVIDIA

https://www.nvidia.com/fr-fr/geforce/drivers/

NOTA : Version de driver Nvidia minimum pour Cuba 11.8 : 452.39

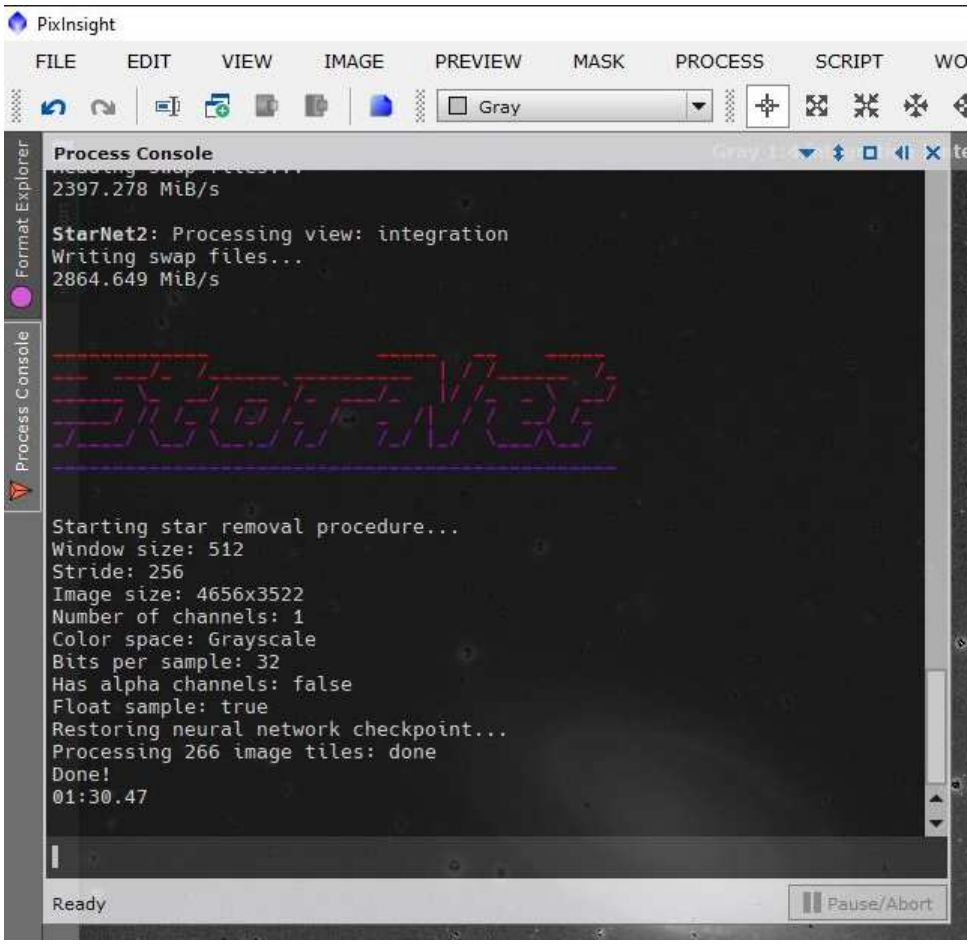
https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html

4 TEMPS DE TRAITEMENT AVANT MODIFICATION

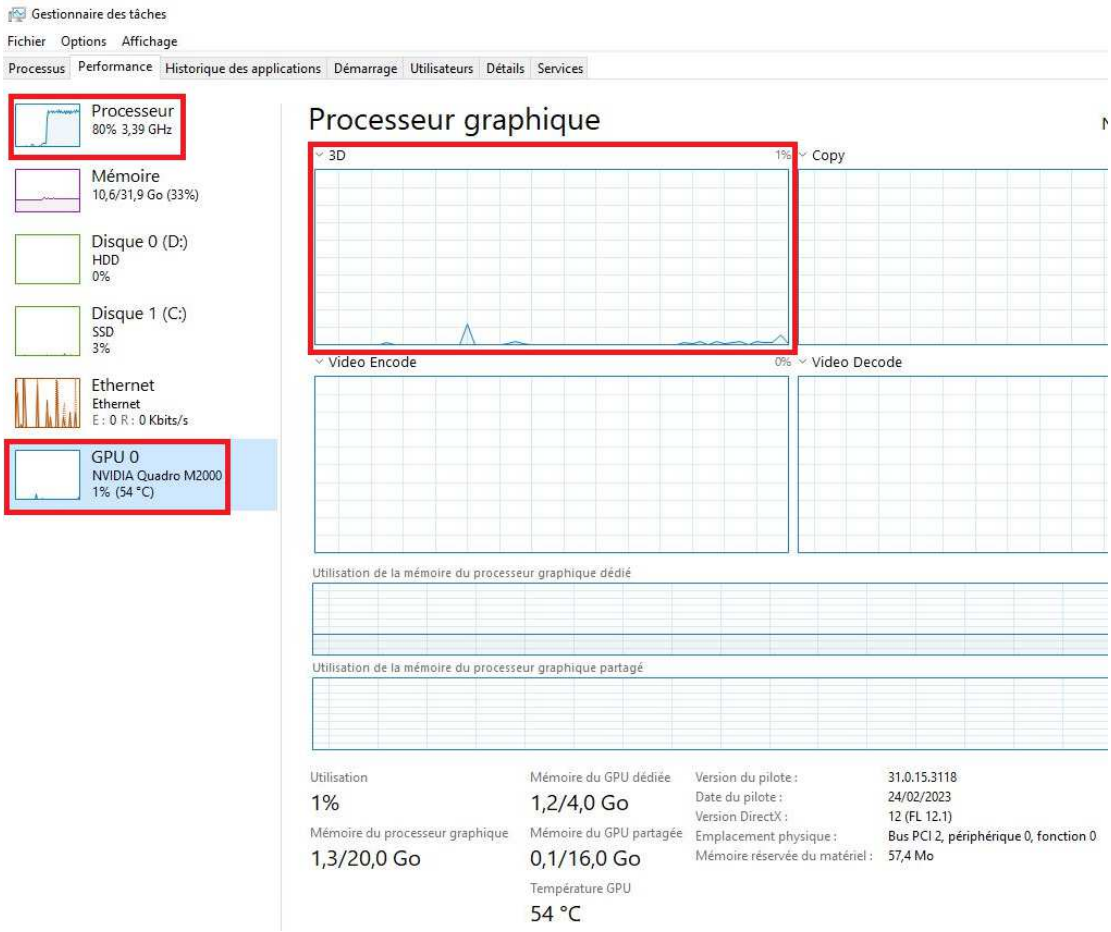
Pour se rendre compte du gain en terme de temps, faire un STARNET2 sur une image

Dans mon cas et avec mon PC, c'est 1mn30…

RDV à la fin du tuto pour voir l'amélioration….



Si on regarde le gestionnaire de tache, onglet performance, on constate que le processeur tourne à 80% et que celui de la carte graphique est à 1%...



5 LOGICIELS A TELECHARGER

Il y a 4 logiciels à télécharger :

- CUDA Toolkit
- cuDNN
- Tensorflow
- Zlib DLL

Nota : pour CUDA prendre la version 11.8. Apparemment avec la version 12, ça ne fonctionne pas (pas testé).

5.1 CUDA Toolkit 11.8

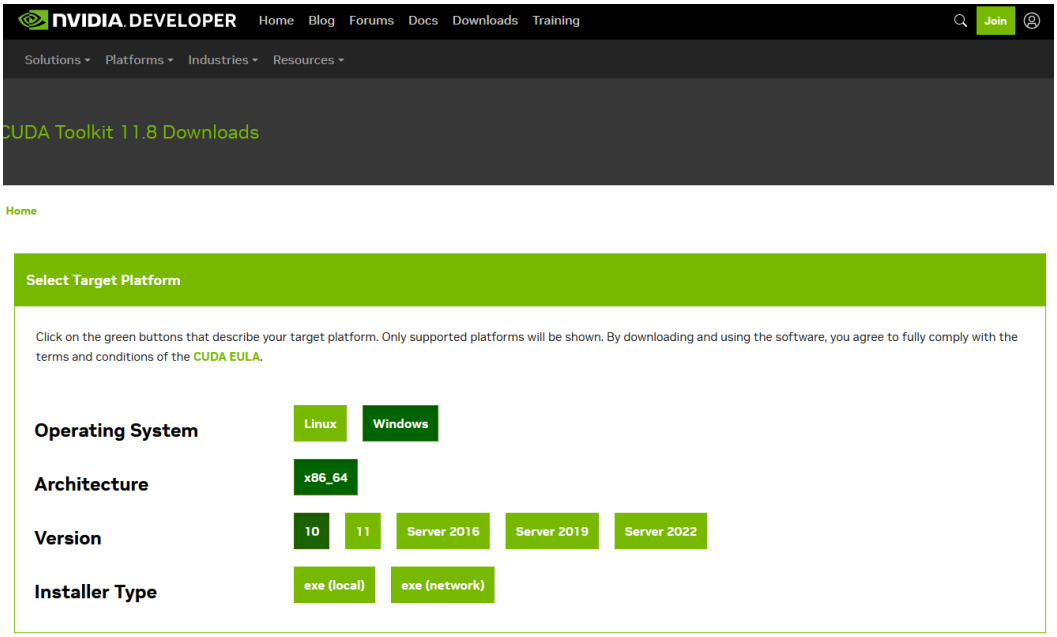
https://developer.nvidia.com/cuda-11-8-0-download-archive?target_os=Windows&target_arch=x86_64&target_version=10

Choisir entre W10 et W11 même si apparemment c'est le meme fichier.

Prendre soit :

- La version NETWORK de 29mo qui nécessitera d'etre connecté à internet lors de l'installation de CUBA pour télécharger des compléments
- La version LOCAL de 3Gb qui doit tout contenir

J’ai pris la version NETWORK.

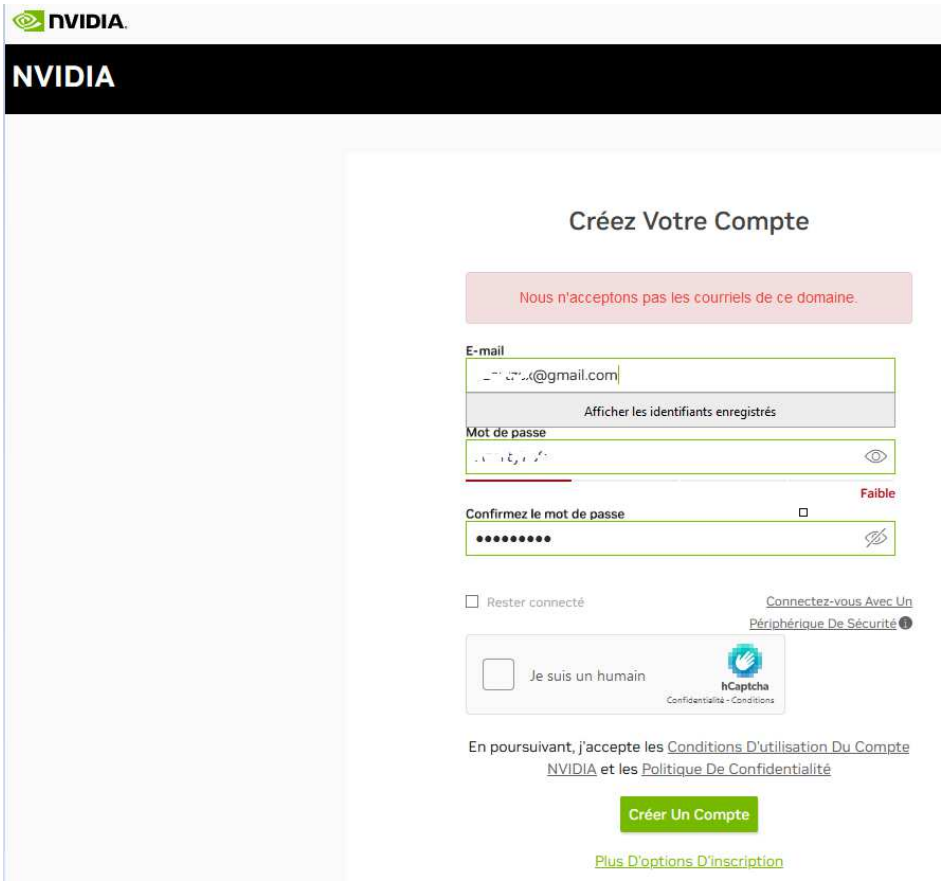


5.2 Cudnn

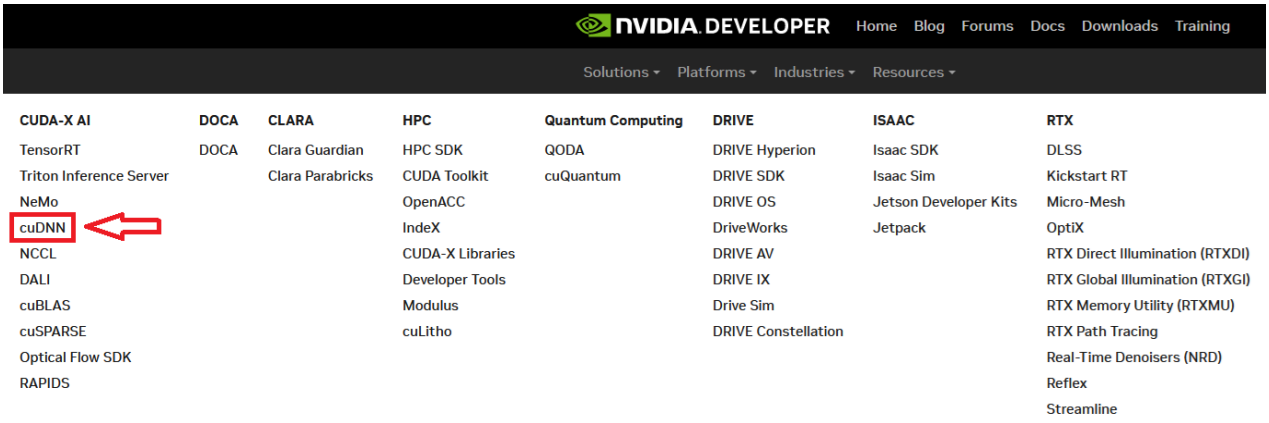
Il faut se créer un compte sur https://developer.nvidia.com/

Si vous ne souhaitez pas indiquer votre mail habituel, se créer un mail bidon sous Gmail

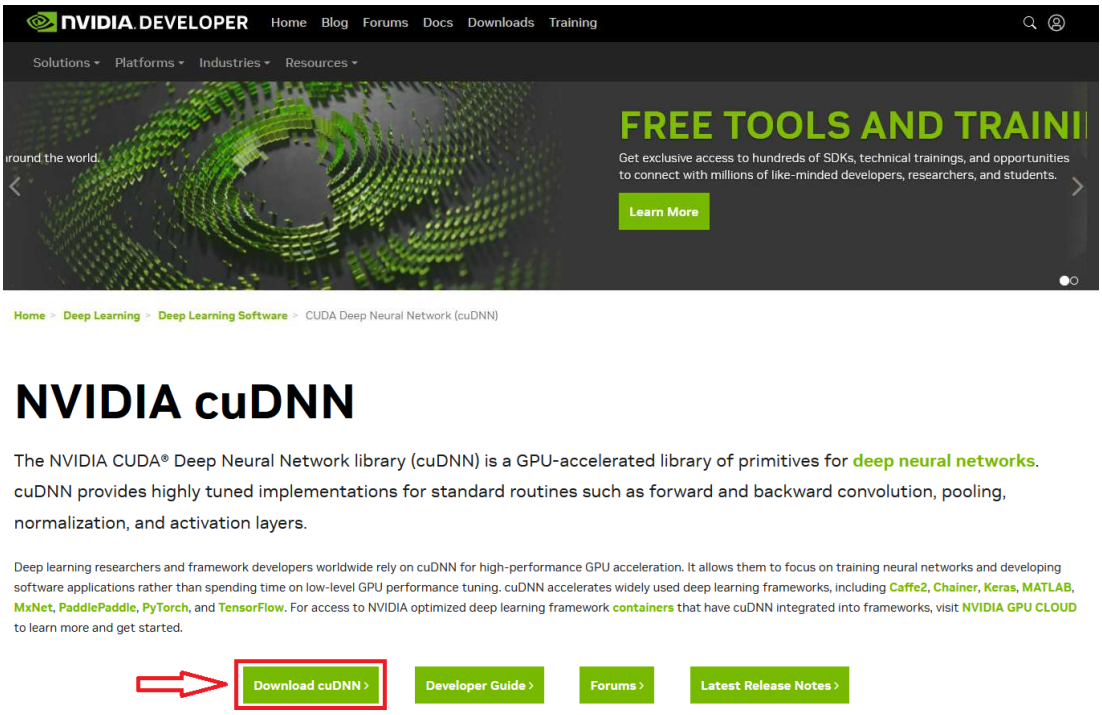
Inutile d’utiliser un mail « temporaire / jetable » comme certains sites le proposent, ça ne fonctionne pas !



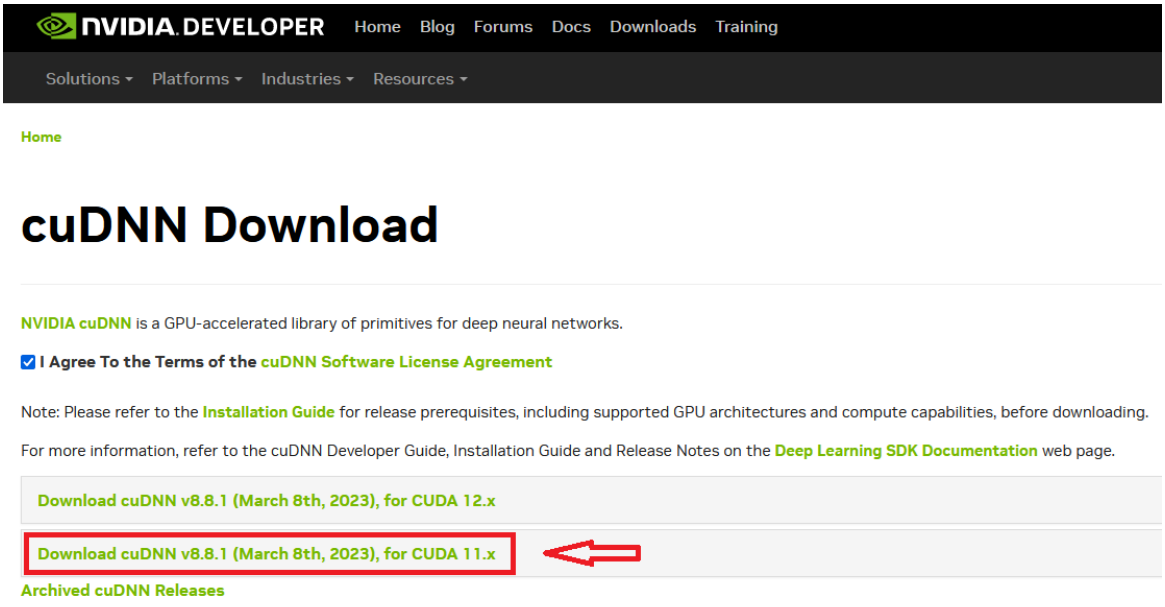
Une fois connecté, aller dans PLATFORMS puis cuDNN



Puis « DOWNLOAD cuDNN »



Puis “ DOWNLOAD cuDNN v8.8.1 for CUDA 11.x ”



5.3 Tensorflow

https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow-gpu-windows-x86_64-2.10.1.zip

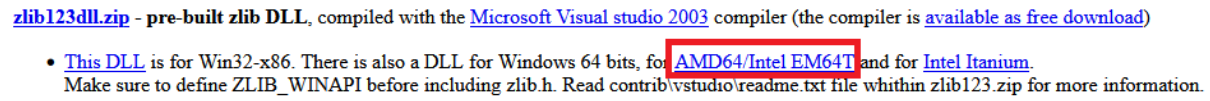
NOTA : Pas réussi à trouvé par le site https://www.tensorflow.org/?hl=fr

Dans les tutos on trouve le lien vers la 2.9.0., j'ai modifié le lien de téléchargement qui se finit par 2.9.0 par 2.9.1 et cela a téléchargé la version 2.9.1. J'ai alors modifié les numéros des versions dans le lien 2.9.2 puis 2.9.3 jusqu'à 2.10.1. En essayant 2.10.2 pas de téléchargement.

5.4 Zlib DLL

Lien direct https://www.winimage.com/zLibDll/zlib123dllx64.zip

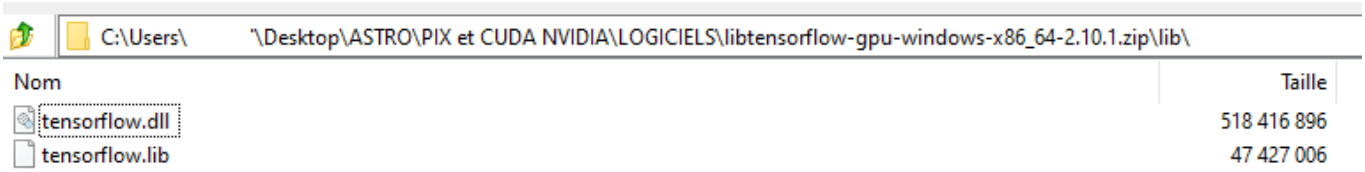
Sinon pour voir si une version plus récente existe, aller sur : https://www.winimage.com/zLibDll/



6 INSTALLATION ET CONFIGURATION

6.1 Étape 1 : Remplacer le tensorflow.dll

- Ouvrir le fichier « libtensorflow-gpu-windows-x86_64-2.10.1.zip »
- Extraire le fichier « tensorflow.dll » qui se trouve dans le répertoire « lib »

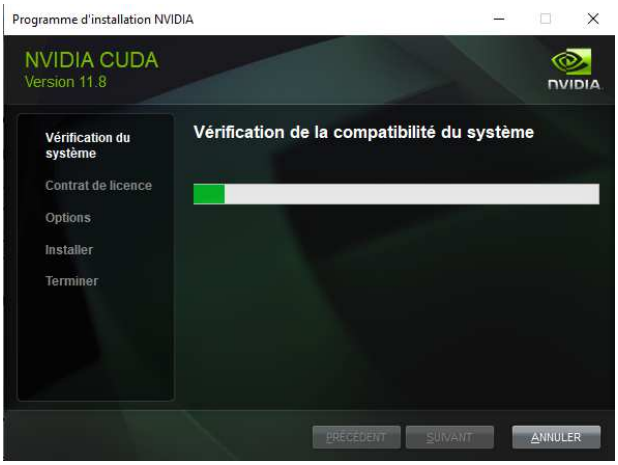


- Ouvrir C:\Program\PixInsight\bin
- Renommer le fichier existant tensorflow.dll en tensorflow-old.dll
- Coller le fichier tensorflow.dll précédemment télécharger

Nota : renommer le fichier en -old permet de réutiliser ce fichier (SANS le -OLD) pour tester avant et apres en termes de temps de traitement.

6.2 Étape 2 : Installer CUDA

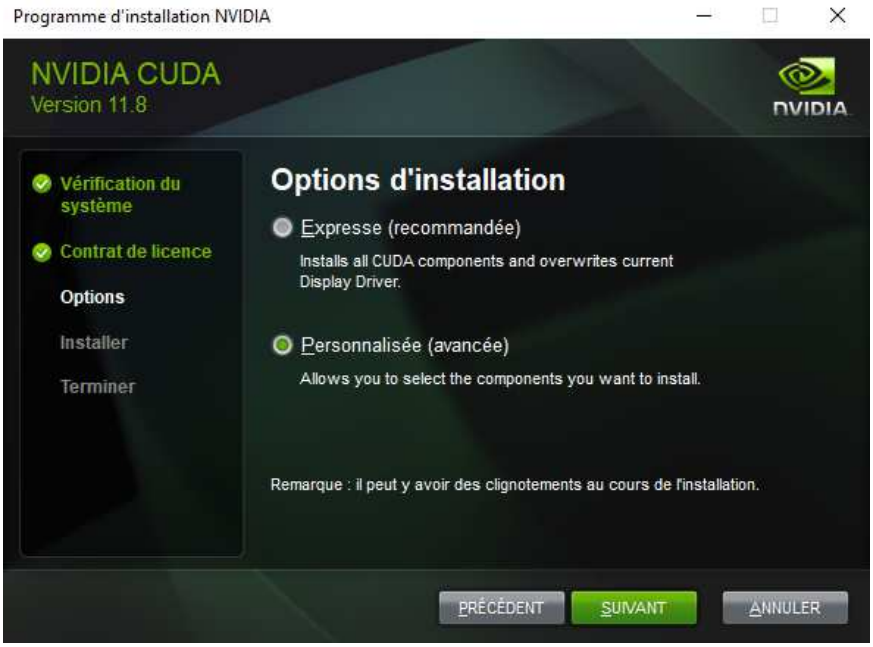
- Lancer l'installation de « cuda_11.8.0_windows_network.exe ».
- Il vérifie la compatibilité du système



- On accepte le contrat de licence



- On sélectionne installation personnalisée

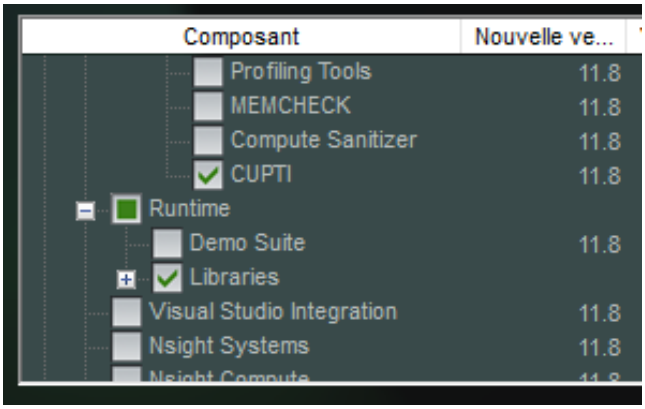


- On décoche tout

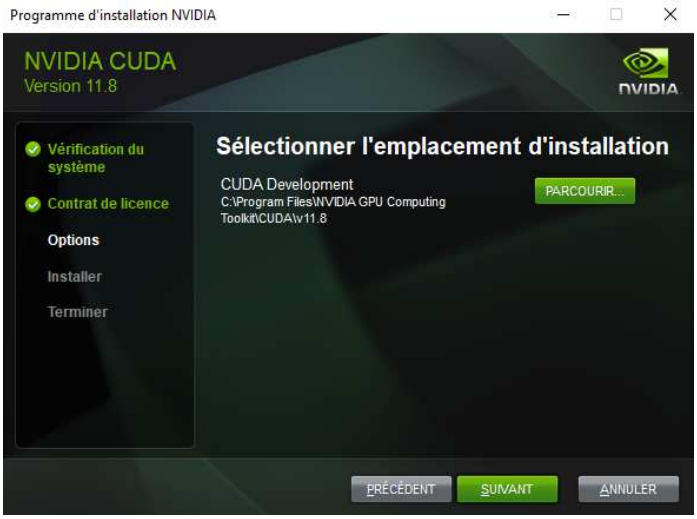


- On coche uniquement
    - CUDA -> Runtime -> Libraries
    - CUDA > Development > Tools > CUPTI





- On valide l'emplacement d'installation

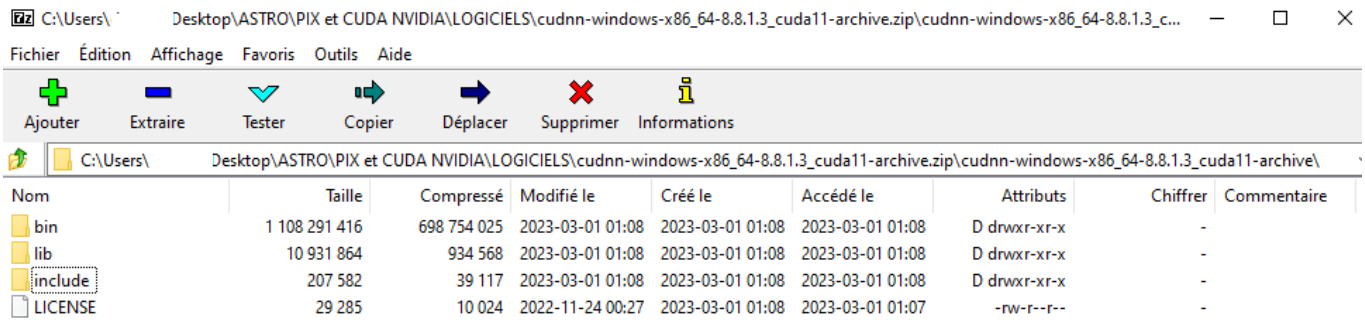


- Ca s'installe et on ferme



6.3 Étape 3 : Copier/coller fichiers cuDNN

- Dézipper le fichier « cudnn-windows-x86_64-8.8.1.3_cuda11-archive.zip »



- Copier tous les fichiers du répertoire « bin » et les coller dans « C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin »

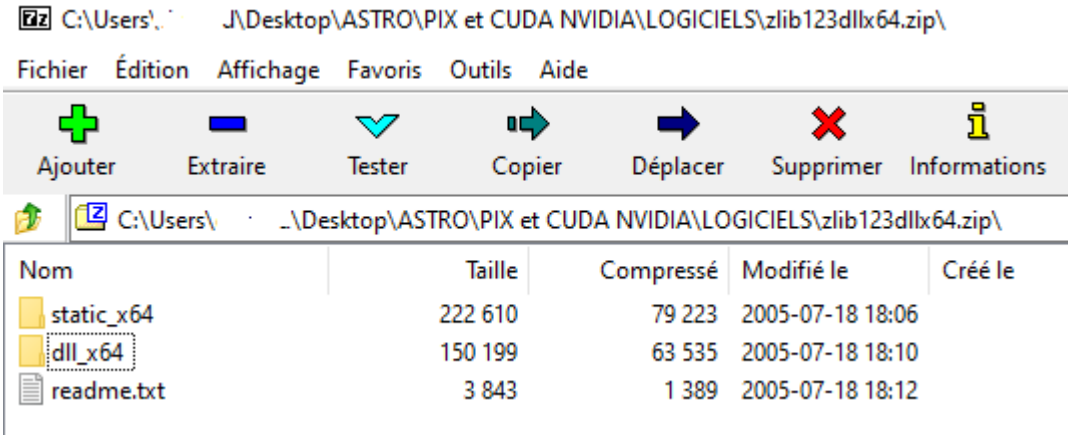
- Copier tous les fichiers du répertoire « lib\x64 » et les coller dans « C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\lib\x64 »

- Copier tous les fichiers du répertoire « include » et les coller dans « C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\include »

6.4 Étape 4 : Copier/coller fichier Zlib

- Ouvrir le fichier « zlib123dllx64.zip »

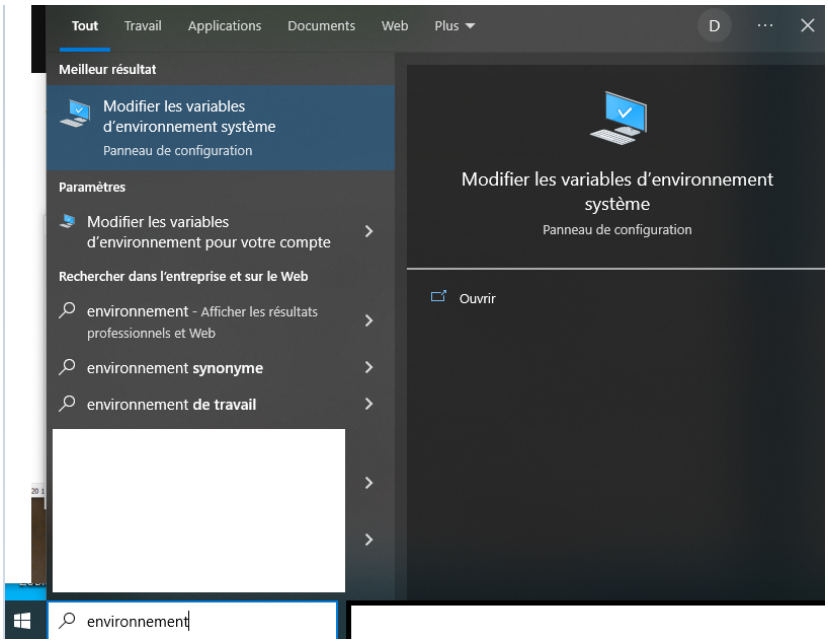
- Extraire le fichier « zlibwapi.dll » qui se trouve dans le répertoire « dll_x64 »



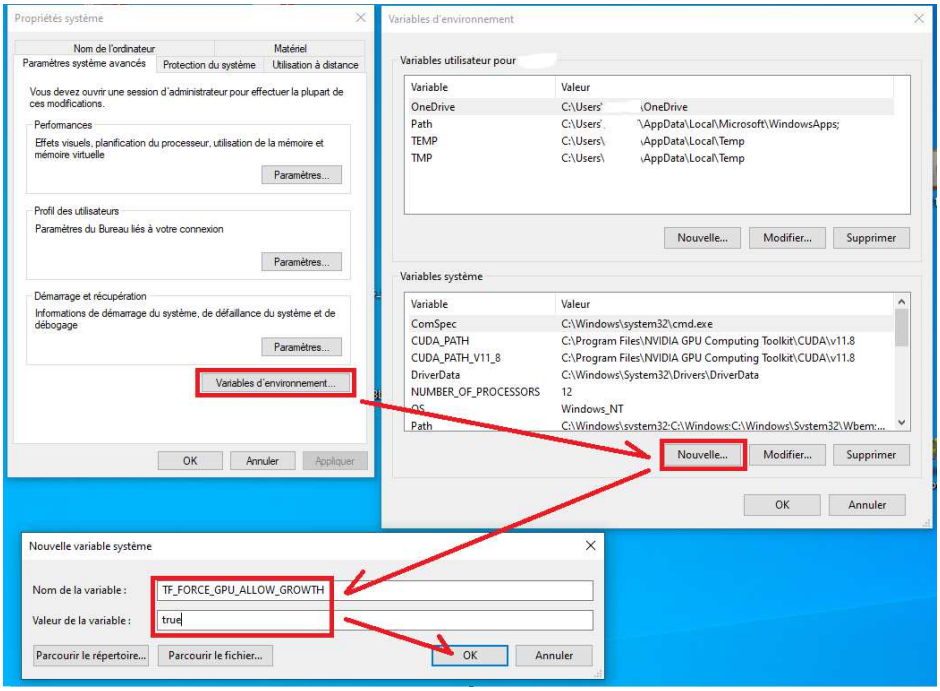
- Le coller dans « C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin »

7 MODIFIER LES VARIABLES D'ENVIRONNEMENT

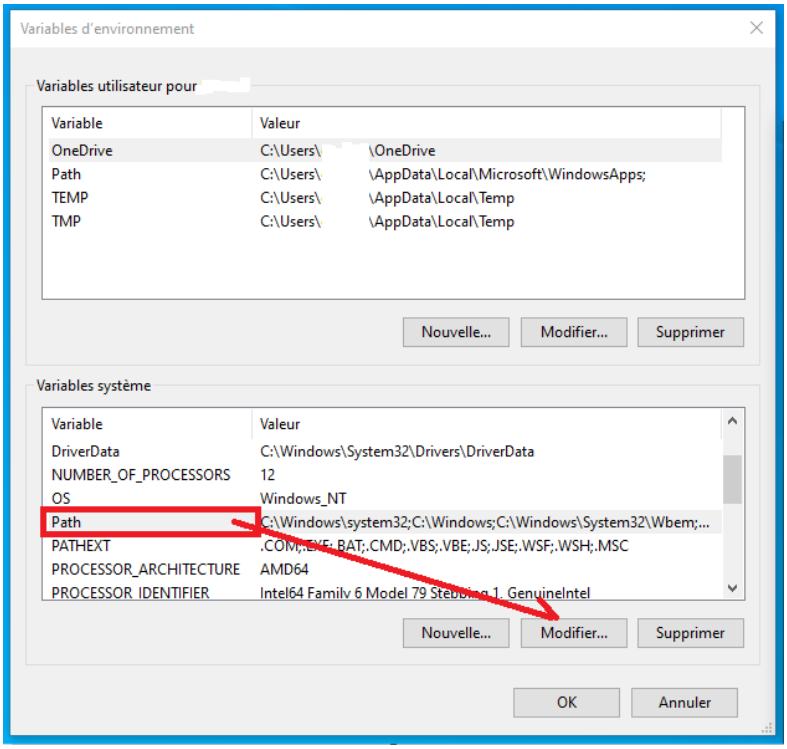
- Dans la barre de recherche de windows, taper « environnement » et ouvrir « Modifier les variables d'environnement système »



- Cliquer sur « Variables d'environnement », puis sous « Variables système » cliquez sur « Nouveau » et entrez « TF_FORCE_GPU_ALLOW_GROWTH » comme nom et « true » comme valeur, puis cliquer sur « OK »



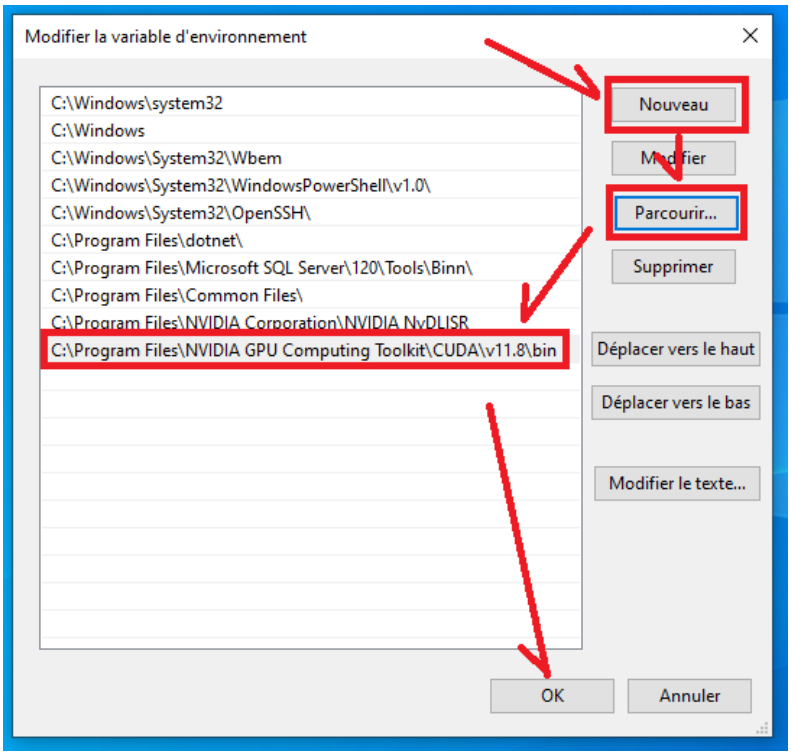
- Sélectionnez la variable appelée « PATH » dans « Variables système », puis cliquez sur « Modifier »



- Cliquer sur « Nouveau » puis « Parcourir »

Aller chercher l'adresse suivante : C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin

Cliquer sur ok et fermer tout.

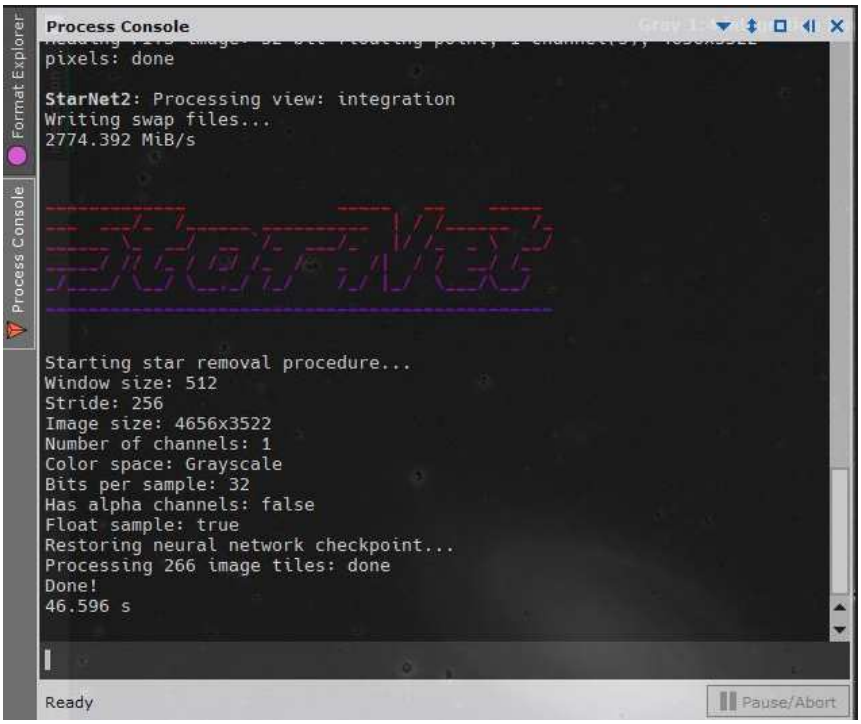


8 RESULTAT

Pour voir le résultat, on refait un STARNET2 sur la même image.

Dans mon cas je suis passé de 1mn30 à 46 secondes, FORMIDABLE !!!

(Un essai avec une RTX 3060 fait descendre le temps à 24 secondes)



Si on regarde le gestionnaire de tache, onglet performance, on constate que le processeur tourne à 12% et que celui de la carte graphique est à quasi 70%

(Pour ma part mais pas le cas pour tout le monde, il fallait me mettre en « compute_0 » pour voir la courbe et on note sur la gauche que le GPU indique 10%...., mais bon l'essentiel est que le temps de traitement diminue)

