

**Recherches bibliographiques sur Yolov3, Yolov4 et Yolov5**

**les protocoles d’installation**

1. **Comment installer YOLOv3**

[**https://pjreddie.com/darknet/install/**](https://pjreddie.com/darknet/install/)

[**https://www.youtube.com/watch?v=-HtiYHpqnBs**](https://www.youtube.com/watch?v=-HtiYHpqnBs)

[**https://www.youtube.com/watch?v=Y73SWT79Rck**](https://www.youtube.com/watch?v=Y73SWT79Rck)

**Installation de Darknet**

Il existe deux options pour installer Darknet :

* OpenCV si vous souhaitez une plus grande variété de types d'images pris en charge.
* CUDA si vous voulez un calcul GPU.
  1. **Installation du système de base**

Cloner le référentiel git Darknet

git clone https://github.com/pjreddie/darknet.git

cd darknet

make

Ensuite, on va voir beaucoup d’informations de compilation

mkdir -p obj

gcc -I/usr/local/cuda/include/ -Wall -Wfatal-errors -Ofast....

gcc -I/usr/local/cuda/include/ -Wall -Wfatal-errors -Ofast....

gcc -I/usr/local/cuda/include/ -Wall -Wfatal-errors -Ofast....

.....

gcc -I/usr/local/cuda/include/ -Wall -Wfatal-errors -Ofast -lm....

Exécutez Darknet

./darknet

Obtenir la sortie

usage: ./darknet <function>

* 1. **Compiler avec CUDA**

Téléchargez et installez CUDA, téléchargez l'adresse du site Web de CUDA: <https://developer.nvidia.com/cuda-downloads>

Après avoir installé CUDA, changez la première ligne du Makefile dans le répertoire de base en:

GPU=1

Vous pouvez maintenant créer le projet et CUDA sera activé. Par défaut, il exécutera le réseau sur la 0ème carte graphique de votre système (si vous avez correctement installé CUDA, vous pouvez lister vos cartes graphiques en utilisant nvidia-smi). Si vous voulez changer la carte que Darknet utilise, vous pouvez lui donner l'indicateur de ligne de commande optionnel -i <index>, comme:

./darknet -i 1 imagenet test cfg/alexnet.cfg alexnet.weights

Si vous avez compilé avec CUDA mais que vous voulez faire des calculs CPU pour une raison quelconque, vous pouvez utiliser -nogpu pour utiliser le CPU à la place:

./darknet -nogpu imagenet test cfg/alexnet.cfg alexnet.weights

* 1. **Compiler avec OpenCV**

Par défaut, Darknet utilise stb\_image.h pour le chargement de l'image. Si vous voulez plus de support pour les formats étranges (comme les jpeg CMJN, merci Obama), vous pouvez utiliser OpenCV à la place! OpenCV vous permet également de visualiser des images et des détections sans avoir à les enregistrer sur le disque. Nous pouvons aller sur le site OpenCV pour en savoir plus, voici son site internet:

<https://opencv.org/superannotate-desktop/>

Tout d'abord, installez OpenCV. Si vous faites cela à partir de la source, ce sera long et complexe, alors essayez de demander à un gestionnaire de paquets de le faire pour vous.

Ensuite, changez la 2ème ligne du Makefile pour lire:

OPENCV=1

Essayez d'abord de refaire le projet. Utilisez ensuite la routine imtest pour tester le chargement et l'affichage de l'image:

./darknet imtest data/eagle.jpg

Voir la fenêtre suivante pour prouver le succès.

鹰站在地上

描述已自动生成

1. **Comment installer YOLOv4**

<https://medium.com/@riteshkanjee/yolov4-tutorial-1-prerequisites-for-yolov4-installation-in-10-steps-6e7ef571cf4b>

https://pypi.org/project/yolov4/#description

10 étapes pour installer YOLOv4 :

* + - * + Installez Python
        + Installation de Git
        + Installation de CMake
        + Installation de Visual Studio
        + Mise à jour du pilote GPU
        + Installation de CUDA
        + Installation de CuDNN
        + Installation d'OpenCV
        + Configuration de CMake OpenCV
        + Construire OpenCV dans Visual Studio

Carte routière

* Tut 1 qui est celui-ci, on passe à mettre en place les pré-requis
* Tut 2, qui est la prochaine conférence, nous allons installer Darknet et implémenter YOLOv4 sur une image
* Tut 3 nous prenons YOLOv4 en détection d'objets en temps réel sur vidéo, pour le traitement et l'enregistrement d'une vidéo et la détection sur une webcam.
* Tut 4, nous créons une application de distanciation sociale en utilisant YOLOv4.

Pour le processus d'installation détaillé, vous pouvez regarder la vidéo à l'adresse du lien : <https://www.youtube.com/watch?v=5pYh1rFnNZs&feature=youtu.be>

**Download Weights**

* [yolov4-tiny.conv.29](https://drive.google.com/file/d/1WtOuGfUgNyNfALo5_VhQ1kb5QenRE0Gt/view?usp=sharing)
* [yolov4-tiny.weights](https://drive.google.com/file/d/1GJwGiR7rizY_19c_czuLN8p31BwkhWY5/view?usp=sharing)
* [yolov4-tiny-relu.weigths(incomplete)](https://drive.google.com/file/d/1K1Nh9j0K-Bj4w2qa_9cE0NrK9vz6BhOF/view?usp=sharing)
* [yolov4.conv.137](https://drive.google.com/file/d/1li1pUtqpXj_-ZXxA8wJq-nzW8h2HWsrP/view?usp=sharing)
* [yolov4.weights](https://drive.google.com/file/d/15P4cYyZ2Sd876HKAEWSmeRdFl_j-0upi/view?usp=sharing)
* [coco.names](https://github.com/hhk7734/tensorflow-yolov4/tree/master/test/dataset)

1. **Comment installer YOLOv5**

[**https://medium.com/@michaelohanu/yolov5-tutorial-75207a19a3aa**](https://medium.com/@michaelohanu/yolov5-tutorial-75207a19a3aa)

[**https://github.com/ultralytics/yolov5**](https://github.com/ultralytics/yolov5)

[**https://www.reddit.com/r/computervision/comments/hk4m5e/install\_run\_yolov5\_object\_detection\_in\_3\_mins/**](https://www.reddit.com/r/computervision/comments/hk4m5e/install_run_yolov5_object_detection_in_3_mins/)

* 1. **Configuration environnementale et installation des dépendances YOLOv5**

D'abord cloner le référentiel YOLOv5 et installer les dépendances. Cela configurera notre environnement de programmation pour qu'il soit prêt à exécuter des commandes d'entraînement et d'inférence de détection d'objets.

Exigences: Python 3.7 ou version ultérieure avec toutes les dépendances requirements.txt installées, y compris torch> = 1.5.

Pour installer YOLOv5 et ses dépendances, exécutez :

$ git clone https://github.com/ultralytics/yolov5 # clone repo

$ pip install -U -r yolov5/requirements.txt # install dependencies

$ cd /content/yolov5 #change directory into project folder.

* 1. **Installer PyTorch localement**

<https://pytorch.org/get-started/locally/>

Sélectionnez vos préférences et exécutez la commande d'installation. Stable représente la version la plus testée et prise en charge de PyTorch. Cela devrait convenir à de nombreux utilisateurs. L'aperçu est disponible si vous voulez la dernière version 1.7, non entièrement testée et prise en charge, qui est générée chaque nuit. Veuillez-vous assurer que vous remplissez les conditions ci-dessous (par exemple, numpy), en fonction de votre gestionnaire de packages. Anaconda est notre gestionnaire de paquets recommandé car il installe toutes les dépendances. Vous pouvez également installer les versions précédentes de PyTorch. Notez que LibTorch n'est disponible que pour C ++.

* + 1. Installation sous Linux

图表, 条形图

描述已自动生成

Python : Python 3.6 ou supérieur est généralement installé

Installer les binaires PyTorch : Anaconda ou pip

**Package Manager et INSTALLATION**

* Anaconda

Sans CUDA

installer PyTorch via Anaconda, et ne pas avoir de système compatible CUDA ou ne pas nécessiter CUDA, dans le sélecteur ci-dessus, choisissez OS: Linux, Package: Conda et CUDA: None. Ensuite, exécutez la commande qui vous est présentée :

pip install torch==1.6.0+cpu torchvision==0.7.0+cpu -f https://download.pytorch.org/whl/torch\_stable.html

Avec CUDA

Pour installer PyTorch via Anaconda, et que vous avez un système compatible CUDA, dans le sélecteur ci-dessus, choisissez OS: Linux, Package: Conda et la version CUDA adaptée à votre machine. Souvent, la dernière version de CUDA est meilleure. Ensuite, exécutez la commande qui vous est présentée(exemple avec CUDA 10.1) :

pip install torch==1.6.0+cu101 torchvision==0.7.0+cu101 -f https://download.pytorch.org/whl/torch\_stable.html

* Pip

Alors que Python 3.x est installé par défaut sur Linux, pip n'est pas installé par défaut. Pour installer pip sur Linux :

sudo apt install python3-pip

Sans CUDA

Pour installer PyTorch via pip, et ne pas avoir de système compatible CUDA ou ne pas nécessiter CUDA, dans le sélecteur ci-dessus, choisissez OS: Linux, Package: Pip et CUDA: None. Ensuite, exécutez la commande qui vous est présentée.

Avec CUDA

Pour installer PyTorch via pip et disposer d'un système compatible CUDA, dans le sélecteur ci-dessus, choisissez OS: Linux, Package: Pip et la version CUDA adaptée à votre machine. Souvent, la dernière version de CUDA est meilleure. Ensuite, exécutez la commande qui vous est présentée.

* + 1. Installation sous macOS

图表, 条形图

描述已自动生成

PyTorch est pris en charge sur macOS 10.10 (Yosemite) ou supérieur.

Il est recommandé d'utiliser Python 3.5 ou supérieur

Installer les binaires PyTorch : Anaconda ou pip

**Package Manager et INSTALLATION**

* Anaconda

Pour installer PyTorch via Anaconda :

conda install pytorch torchvision -c pytorc

* Pip

Si vous avez installé Python via Homebrew ou le site Web Python, pip a été installé avec. Si vous avez installé Python 3.x, vous utiliserez la commande pip3.

Pour installer PyTorch via pip :

# Python 3.x

pip3 install torch torchvision

* + 1. Installation sous Windows

Windows 7 et supérieur; Windows 10 ou supérieur recommandé.

Windows Server 2008 r2 et supérieur

PyTorch sous Windows ne prend en charge que Python 3.x

Installez Python : chocolatey, Python website, Anaconda

Installer les binaires PyTorch : Anaconda ou pip

**Package Manager et INSTALLATION**

* Anaconda

Pour installer Anaconda, vous utiliserez le programme d'installation graphique 64 bits pour PyTorch 3.x. Cliquez sur le lien du programme d'installation et sélectionnez Exécuter. Anaconda sera téléchargé et l'invite du programme d'installation vous sera présentée. Les options par défaut sont généralement saines.

Pour installer PyTorch avec Anaconda, vous devrez ouvrir une invite Anaconda via

Start | Anaconda3 | Anaconda Prompt.

* Pip

Si vous avez installé Python par l'une des méthodes recommandées ci-dessus [LINK], [pip] (https://pypi.org/project/pip/) aura déjà été installé pour vous.

**3.3 VÉRIFICATION**

Pour nous assurer que PyTorch a été correctement installé, nous pouvons vérifier l'installation en exécutant un exemple de code PyTorch. Ici, nous allons construire un tenseur initialisé aléatoirement.

from \_\_future\_\_ import print\_function

import torch

x = torch.rand(5, 3)

print(x)

La sortie devrait être quelque chose de similaire à:

tensor([[0.3380, 0.3845, 0.3217],

[0.8337, 0.9050, 0.2650],

[0.2979, 0.7141, 0.9069],

[0.1449, 0.1132, 0.1375],

[0.4675, 0.3947, 0.1426]])

Vérifiez si votre pilote GPU et CUDA sont activés et accessibles par PyTorch

import torch

torch.cuda.is\_available()

1. **Comment installer YOLOv3 sur Jetson TX2**

[**https://www.ncnynl.com/archives/201909/3369.html**](https://www.ncnynl.com/archives/201909/3369.html)

* 1. **Télécharger darknet**

mkdir -p ~/dl/darknet

cd ~/dl/darknet

git clone https://github.com/pjreddie/darknet.git

cd darknet

* 1. **Modifiez le fichier Makefile, Activer le GPU et opencv**

GPU=1

CUDNN=1

OPENCV=1

OPENMP=0

DEBUG=0

ARCH= -gencode arch=compute\_53,code=[sm\_53,compute\_53] \

-gencode arch=compute\_62,code=[sm\_62,compute\_62]

# -gencode arch=compute\_20,code=[sm\_20,sm\_21] \ This one is deprecated?

# This is what I use, uncomment if you know your arch and want to specify

# ARCH= -gencode arch=compute\_52,code=compute\_52

* 1. **Compiler**

make -j4

* 1. **Juge réussi**

./darknet

usage: ./darknet <function>

* 1. **Installer PyTorch sur Jetson TX2**

[**https://forums.developer.nvidia.com/t/pytorch-for-jetson-version-1-6-0-now-available/72048**](https://forums.developer.nvidia.com/t/pytorch-for-jetson-version-1-6-0-now-available/72048)

* + 1. PyTorch pip wheels

Puisque la carte Jetson TX2 est basé sur l’architecture ARM, donc il faut utiliser les fichers wheel correspondant . Ci-dessus un exemple pour installer Pytorch v1.6.0 sur Jetson TX2 avec JetPack 4.4 installé, Python3.6.

* + 1. Instructions

Les commandes d’installlation sur Jetson TX2 :

Wget https://nvidia.box.com/shared/static/9eptse6jyly1ggt9axbja2yrmj6pbarc.whl -O torch-1.6.0-cp36-cp36m-linux\_aarch64.whl

sudo apt-get install python3-pip libopenblas-base libopenmpi-dev

pip3 install Cython

pip3 install numpy torch-1.6.0-cp36-cp36m-linux\_aarch64.whl

L’installation de torchvision

$ sudo apt-get install libjpeg-dev zlib1g-dev

$ git clone --branch <version> https://github.com/pytorch/vision torchvision # see below for version of torchvision to download

$ cd torchvision

$ export BUILD\_VERSION=0.x.0 # where 0.x.0 is the torchvision version

$ sudo python setup.py install # use python3 if installing for

Pour vérifier l’installation de Pytorch, lancer les commandes ci-dessus dans Python3.6

>>> import torch

>>> print(torch.\_\_version\_\_)

>>> print('CUDA available: ' + str(torch.cuda.is\_available()))

>>> print('cuDNN version: ' + str(torch.backends.cudnn.version()))

>>> a = torch.cuda.FloatTensor(2).zero\_()

>>> print('Tensor a = ' + str(a))

>>> b = torch.randn(2).cuda()

>>> print('Tensor b = ' + str(b))

>>> c = a + b

>>> print('Tensor c = ' + str(c))

>>> import torchvision

>>> print(torchvision.\_\_version\_\_)