Spectrum Technology

V1.0 2022/5/16

抜粋版
はじめてのAI転移学習キット
nvidia tao toolkitを使ってAI 開発スピードを 10 倍に加速
学習編（TAO版）

スペクトラム・テクノロジー株式会社

https://spectrum-tech.co.jp

sales@spectrum-tech.co.jp

学習キット　目次

抜粋版のため目次は一致しません

学習キット　目次

抜粋版のため目次は一致しません

Ubuntu運用マニュアル

1. Ubuntuについて

   Linuxの中でも一番シェアの高いOSです。2004年にDebian系から派生。

2. Linux基本コマンド

   ① システム関係

   - 起動：電源を入れると自動で起動します。
   - 再起動：$ reboot

     又は、左上のメニューの「ゲストを再起動
   - 終了： $ shutdown

     又は、左上のメニューの「ゲストをシャットダウン」
   - ログアウト $ exit

     ルートからログアウトします
   - 日本語／英語の入力切替：半角／全角のボタン（ESCボタンの下）

Ubuntu運用マニュアル

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

masa@ubuntu:~$cd /home/masa/Documents　　ディレクトリの切り替え
masa@ubuntu :/home/masa/Documents$ ls　ファイルとディレクトリの表示（表示したら操作したいファイルを右クリックでコピペして操作します
masa@ubuntu:~$ cp ファイル名　ディレクトリ　配下のディレクトリのファイルを別のディレクトリへコピー
masa@ubuntu:~$ mv ファイル名　ディレクトリ　　配下のディレクトリのファイルを別のディレクトリへ移動
masa@ubuntu:~$ rm ファイル名　　ファイルの削除
便利な機能　　rm –help　　コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

③ ユーザ権限、プロセス他

masa@ubuntu :~ $ su -　　スーパーユーザ（root)に切り替え、パスワードを入力
masa@Ubuntu:~$ sudo　　ルート権限で各種コマンドを実施します。
masa@ubuntu:~$ ps　a　　現状の動いているプロセスを表示
masa@ubuntu:~$ kill　　特定のプロセスを強制終了
masa@ubuntu:~$ apt-get install pkg　　パッケージのインストールなどに使用
masa@ubuntu:~$ date　　日付、時間の設定を行います。
masa@ubuntu:~$ leafpad /etc/network/interfaces　　インタフェースに記述してい内容を変更します。Viよりも使いやすいです。

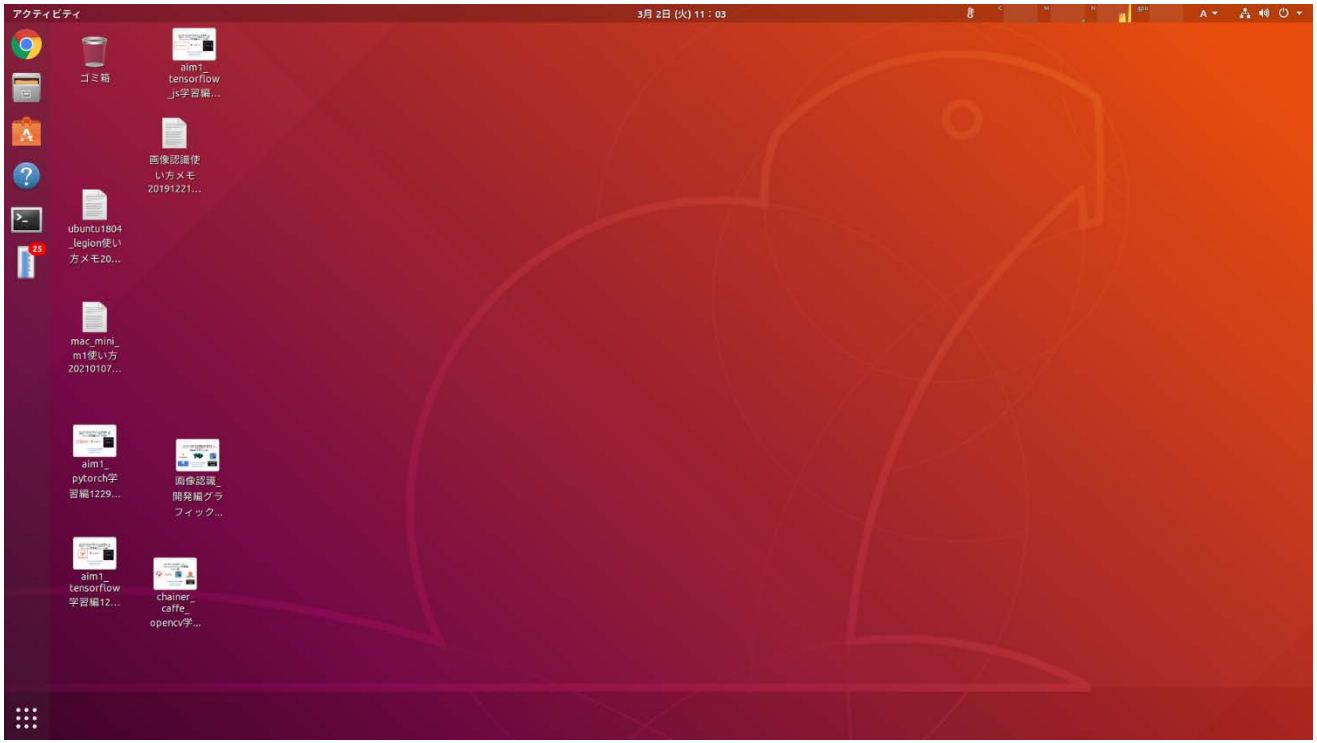
④ モジュール、usb、メモリ、HDDなどの表示

masa@ubuntu:~$ lsmod　　linuxのモジュールリスト表示
masa@ubuntu:~$ lsusb　　usbのデバイス表示
masa@ubuntu:~$ free –mt　　メモリ使用状態表示
masa@ubuntu:~$ df　　HDD（マイクロSD）の使用状態表示

Ubuntu運用マニュアル

3. 基本操作

① 表示画面と内容



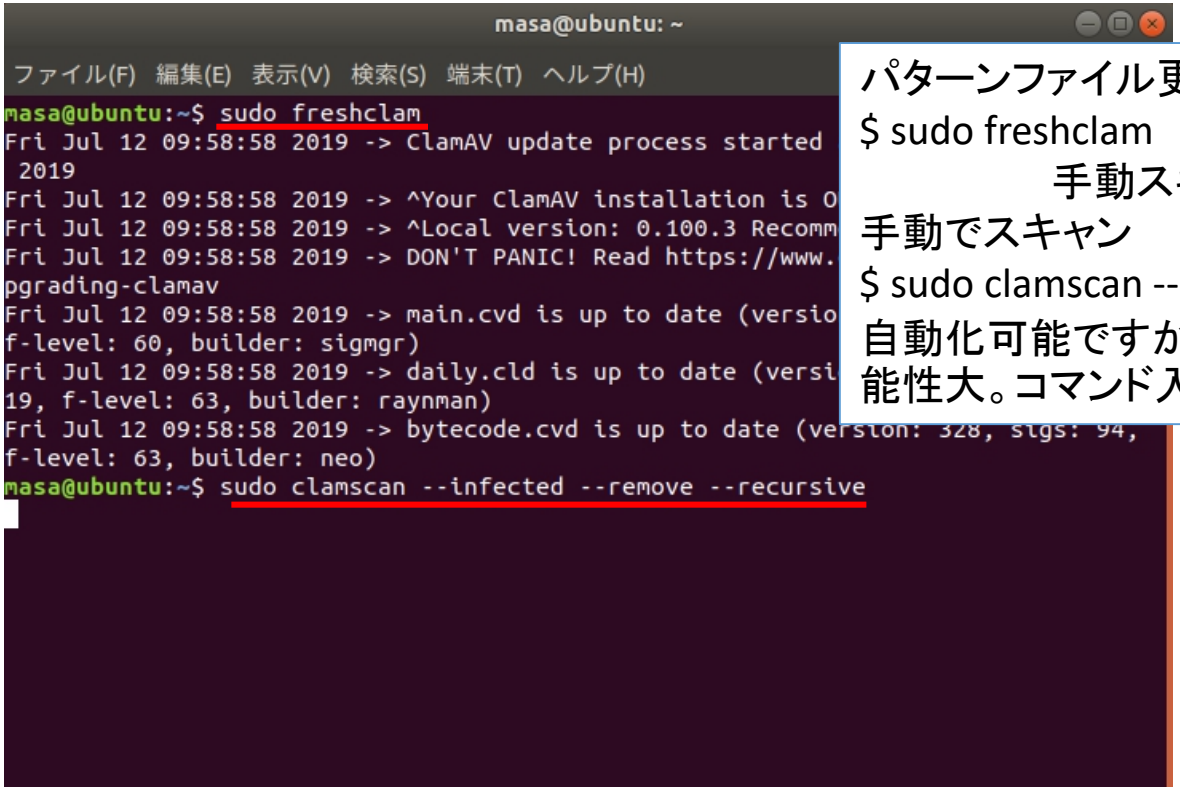
主に使用するもの
・ブラウザ：Chrome
・フォルダ：Documents内に必要なファイルがあります。
・コマンド：コマンド画面を立ち上げて、python3のプログラムを動作させます。

Ubuntu運用マニュアル

4. 日常運用

① セキュリティ対策（アンチウイルス更新、スキャン）

- アンチウイルス対策として無料のclamAVをインストールしてます。
- 手動での運用を基本としてます。



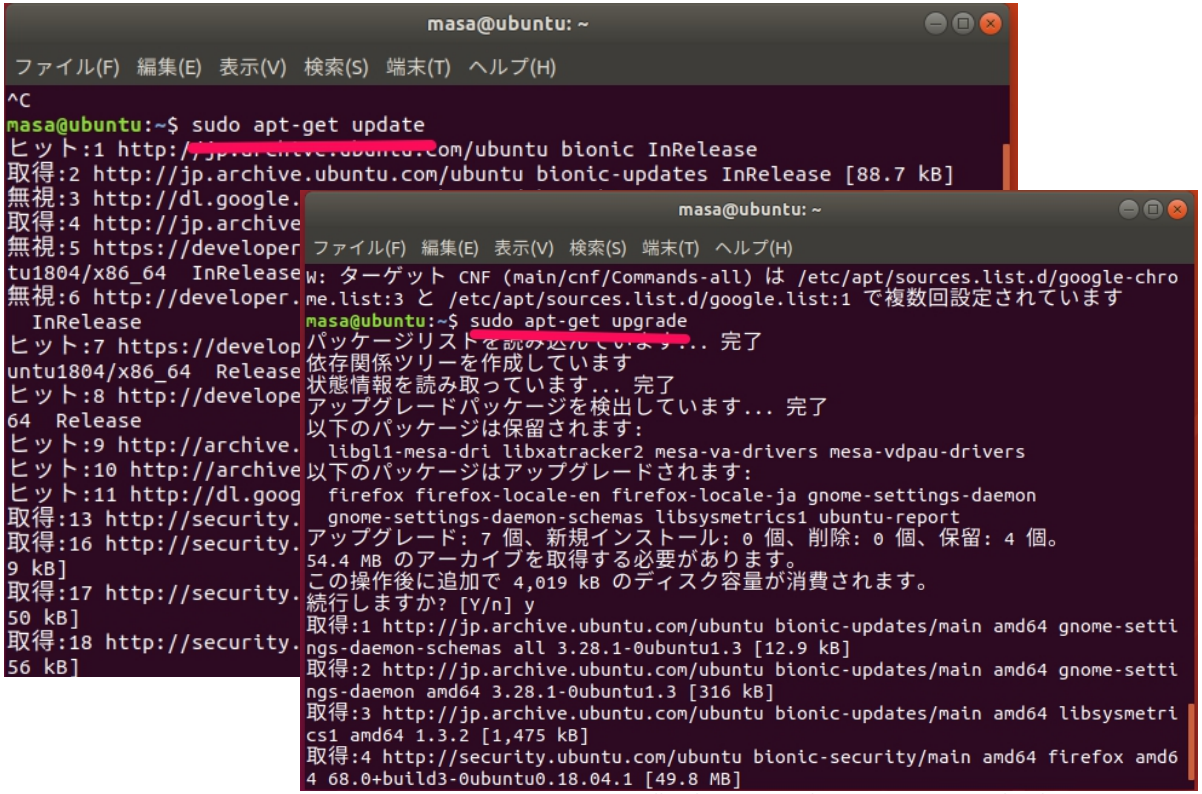
パターンファイル更新
$ sudo freshclam
手動スキャン時にも更新されます
手動でスキャン
$ sudo clamscan --infected --remove --recursive
自動化可能ですが、バックグランドで重くなる可能性大。コマンド入力後時間がかります。

Ubuntu運用マニュアル

4. 日常運用

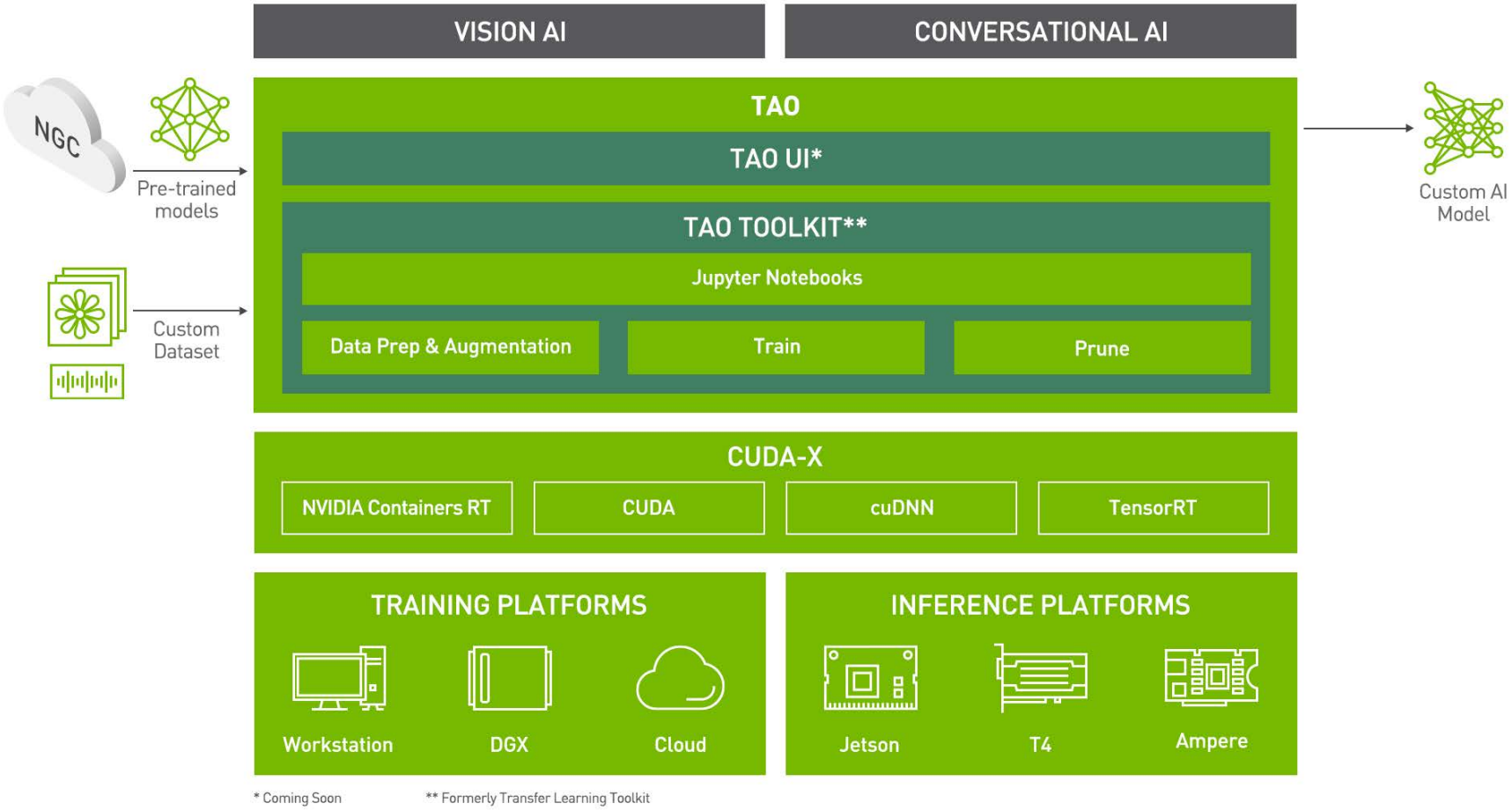
② インストール済パッケージの更新リスト、アップグレード

- Linuxの場合は、頻繁に更新が発生します。アップグレードを定期的に実施してください。
- 更新前には、バックアップを取ることをお勧めします。特にアップグレードはまれに動作不良、戻せない状態が発生します。自己責任で実施してください。



更新リスト取得
$ sudo apt-get update
アップグレード実施
$ sudo apt-get upgrade

学習キット　概念図(TAO版）



Spectrum Technology

学習キット　全体像(TAO版）

ハードウエア

CPU

8 core cpu,32GB RAM

＋

GPU

Nvidia A100, V100, RTX30シリーズに対応

USBメモリ

OS

ubuntu

AI系

TensorRT

TensorFlow

PyTorch

ONNX

ソフトウエア

NVIDIA NGC

NVIDIA TAO

docker

画像系

OpenCV

YOLO

プログラム言語

python

グラヒック

NVIDIA CUDA

Spectrum Technology

ハードウエア概要

①必要なハードウエア仕様

ハードウエアの概要です。

区分		プロダクツ	メーカ	備考
USBメモリ（tao版）		512GB　USB3.0 TAO 関連ソフト		
お客様準備品				
PC本体	cpu	Intel i9, i7, amd Ryzen 9,7など 8 core cpu		GPUが搭載できるものに限ります。
	GPU	A100,　V100 RTX30シリーズ	nvidia	
	メモリ	32GB以上		
	SSD	512GB以上		巨大なdatabaseをインストールするため1TBが望ましい

2. ソフトウエア概要

①ソフトウエアー覧

ソフトウエアの概要です。

区分	ソフト名	バージョン	備考
OS	ubuntu	18.04.3 LTS	
GPU用	cuDNN	8.2.4+cuda11.4	Nvidia用,搭載するGPUに依存 本キットでは、コンテナで使用するため参考
プログラム言語	python3	3.6.9	仮想化で使用
TAO	TAO tool kit	3.0.21.8	
コンテナ関連	Docker-ce	>19.03.5	
	nvidia-container-toolkit	>1.3.0-1	
	Nvidia-driver	>465	
AI用プログラム	tensorRT	8.4	
	tensorflow	1.15	Taoコンテナで使用
	Pytorch	1.8	
	onnx	1.8.1	
各種モジュール	Jupyter notebook、matplotlibなど多数のpipライブラリ		

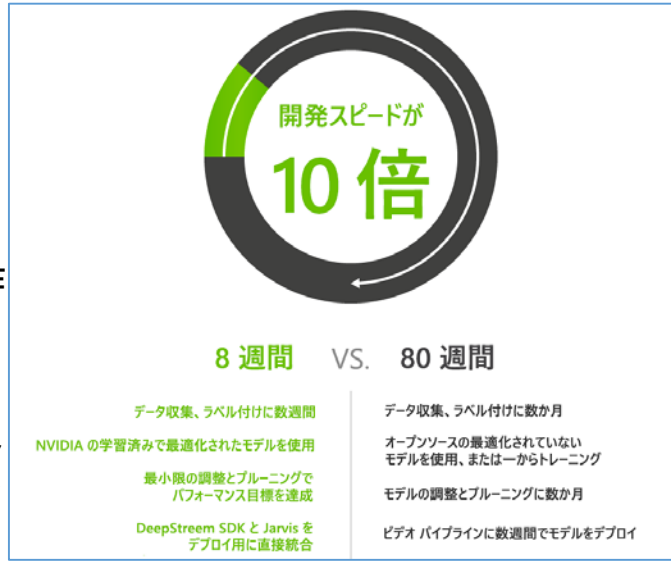
公式サイト
https://developer.nvidia.com/ja-jp/tao-toolkit

Tao toolkit

1. 概要

① 概要

- お金をかけて AI の専門知識を習得しなくても、AI 開発スピードを 10 倍に加速できます。トレーニングをスピードアップして、精度の高い、高性能のドメイン別 AI モデルをすばやく作成できます。
- ビジネスの課題を解決するための AI/ML モデルを一から作成するのは、お金と時間がかかります。転移学習 は、学習済みの特徴量を、既存のニューラル ネットワーク モデルから新しいモデルに抽出するときによく使われる手法です。NVIDIA TAO Toolkit は AI/DL フレームワークの複雑さを抽象化する AI ツールキットです。高品質の学習済みモデルを使用するため、わずかな量のデータでも運用品質のモデルをすばやく構築できます。



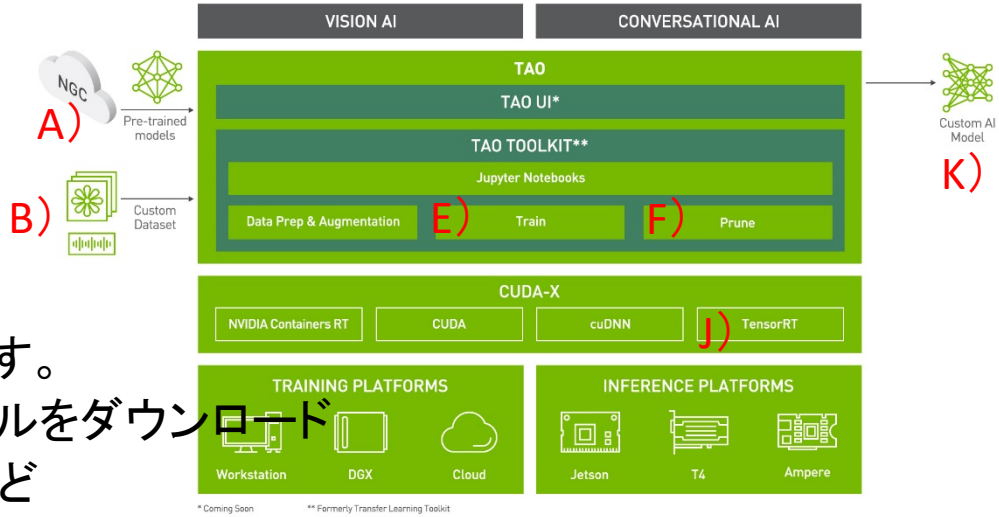
② 利用方法

- TAO を使用すると、NVIDIA の運用品質の学習済みモデルをそのままデプロイするか、コンピューター ビジョンや対話型 AI などのさまざまなユースケースに合わせてモデルを微調整して使用できます。TAO は、AI 作成のための UI ベースのガイド付きワークフロー、TAO (Train, Adapt and Optimize) プラットフォームの中核的なコンポーネントです。
- https://docs.nvidia.com/tao/tao-toolkit/text/overview.html

Tao toolkit

1. 概要

③ 概念図



以下の手順で転移学習を行います。

A) ngcで作成済の学習済みモデルをダウンロード
B) カスタムデータを設定：cocoなど
C) カスタムデータをtensorflow用tfrecordに変換
D) Tfrecordを使って学習
E) 学習したモデルの評価：時間がかかります(モデルによりrtx3080で10時間など）
F) モデルのプルーン化（prune)：切り落とし
G) プルーン化モデルの再学習：時間がかかります
H) 再学習モデルの再評価
I) 推論（inference)
J) モデルのエキスポート（int8最適化、tensorRT変換）
K) エキスポートモデルでの推論

別売：TensorRT学習・開発キットを参照ください

TAO Toolkit

1. 概要

④. アプリー覧

id	区分	アプリ名	概要	tao版	deepstream版 先頭数字は、目次、 二つ目が細分
1	物体検出	DetectNet_V2	車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現	●	●5_7
2		FasterRCNN	同上のデータセット、事前学習モデルで、Faster RCNNモデルで学習する。正解率は、高いが、処理が遅い	●	●3_6
3		SSD	SSD（single shot detector)は、画像中の物体を単一のディープニューラルネットワークで検出する。一般には、認識精度も高く高速といわれているが？結果は、反対。	●	●3_5
4		YOLOV3	darknetで有名なyolo、高速認識が売り	●	●3_7
5		YOLOV4	darknetで有名なyolo、高速認識が売り、新たな開発者が担当	●	
6		yolo_v4_tiny	darknetで有名なyolo、高速認識が売り、新たな開発者が担当、小型モデル版（軽い）	●	
7		RetinaNet	一般的には、2段階で検出していたものを、速度を維持したままで、精度が高い一段階検出モデルができないかと考え、RetinaNetが発表されました	●	
8		efficientdet	efficient detを使った物体認識	●	
9		DSSD	deconvolutional single shot detector(DSSD)	●	
10	セグメンテーション	MaskRCNN	物体のセグメンテーションして、背景をマスクする。	▲エラー	●5_6
11		UNET	医療系の画像を使って、血管などをセグメンテーションする	●	
12	個別事例	action_recognition_net	ビデオから行動認識を行います	▲エラー	▲3_17エラー
13		bpnet	body pose netは、からだの目、耳、手首などの位置を検出します。	●	
14		image_classification	物体の検出します	●	▲5_8エラー
15		emotionnet	感情分類	●	▲5_3エラー
16		facenet	顔検出	●	
17		fpenet	顔の部位検出	●	
18		gazenet	顔から視線を推測	●	▲5_4エラー
19		gesturenet	指の動作を検出	●	▲5_5エラー
20		heartratenet	心電図による推測	▲データ未入手	
21		lprnet	ナンバープレートの文字認識	●	▲4_8エラー
22		multitask_classification	ファッション用品の分類	●	
23		dashcamnet	車種、メーカ名などを識別		●4_3
24		FaceDetectIR	顔識別		●4_4
25		VehicleMakeNet	車種、メーカの認識		●4_5
26		vehicletypenet	車種の認識：クーペ、suv,ban		●4_6
27		PeopleNet-ResNet34	人認識：deepstreamer使用		●4_2
28		TrafficCamNet	車両認識：deepstreamer使用		●4_1
29		LPD	ナンバープレート検出		▲4_7エラー
30		Facial Landmark	顔の細部の位置を認識		●5_2
31		PeopleSegNet	人体セグメンテーション		●5_9
32		People Semantic Segmentation	人体セグメンテーション		●5_10
33		2D Body Pose Estimation	2D人体のポーズ認識		●5_1
34		imagedata-multistream	2画面ストリーミング		●2_12

deepstream版のエラーは、 engine生成の問題。Tao-converterのエラー不明

入力コマンド
$ source /home/masa/venv_py36/bin/activate
$ tao --help

TAO Tool kit
2.事前準備と使い方

① python仮想環境

- 設定編で準備したvenv_py36で全て動作させます。

$ source /home/masa/venv_py36/bin/activate
(venv_py36) masa@ubuntu2 :$

② TAO tool kitの動作方法

- 動作方法には以下の二つがあります。
- https://docs.nvidia.com/tao/tao-toolkit/text/tao_toolkit_quick_start_guide.html

A) Jupyter notebook（推奨）
- 基本的にjupyter notebookで動作させます。全体の流れがわかるので理解しやすいです。

B) Tao cli
- Tao の個別コマンドを使って使用。
- .tao_mounts.jsonを準備してから操作、jupyter notebookで解説しています。

$ tao --help　　ヘルプコマンドで内容を確認
$ tao list　　動作中のtao コンテナを確認できます

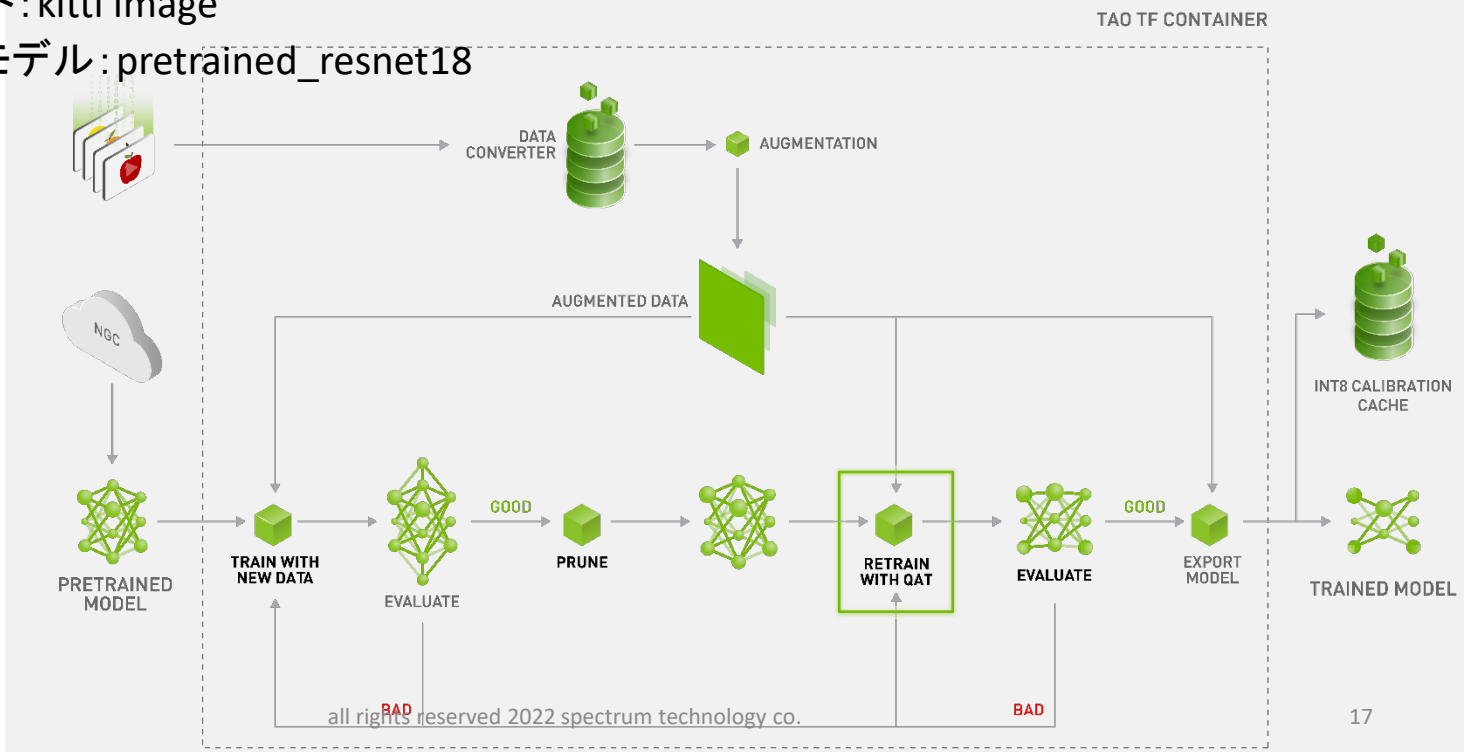
TAO Tool kit

3.事例：物体認識

① detectnet_v2：物体認識

- 概要：車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、detectnet_v2アルゴリズムを使い学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現。
- https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/detectnet_v2.html
- データセット：kitti image
- 事前学習モデル：pretrained_resnet18

全体図



入力コマンド
$ $ cd /home/masa/Documents/tao/cv_samples_v1.3.0/detectnet_v2
$ jupyter notebook

TAO Tool kit

3.事例：物体認識

① detectnet_v2：物体認識

- 概要：車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現。
- https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/detectnet_v2.html
- データセット：kitti image
- 事前学習モデル：pretrained_resnet18

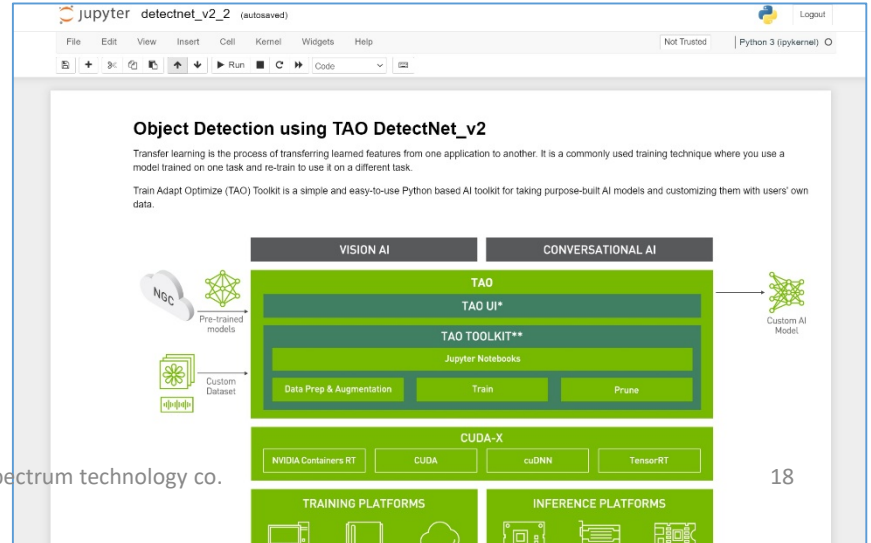
ディレクトリは各自違います

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/detectnet_v2

(venv_py36) masa@ubuntu2 :$ jupyter notebook

detectnet_v2_2.ipynbを選択

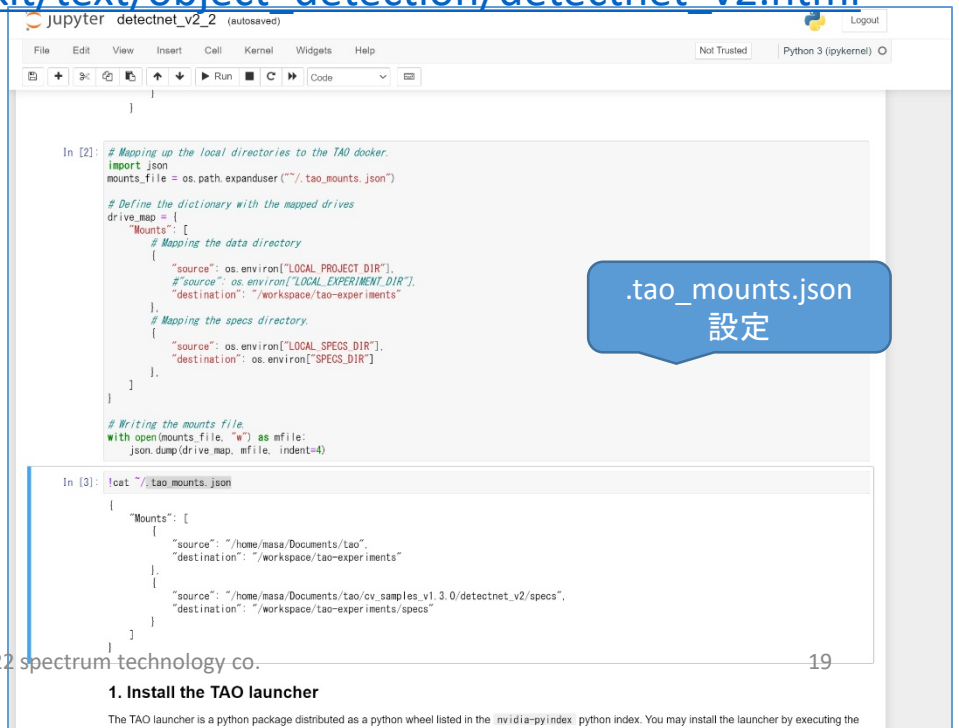
オリジナルは、detectnet_v2.ipynbです。



TAO Tool kit
3.事例：物体認識

① detectnet_v2：物体認識

- 概要：車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、detectnet_v2アルゴリズムを使って学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現。
- https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/detectnet_v2.html
- detectnet_v2_2.ipynbを選択

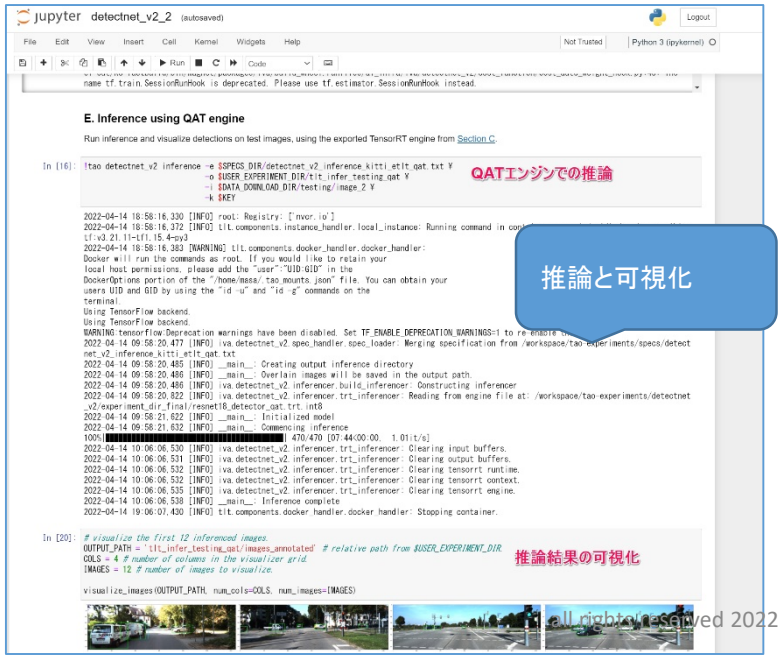


Spectrum Technology

TAO Tool kit
3.事例：物体認識

① detectnet_v2：物体認識

- 概要：車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現。
- https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/detectnet_v2.html
- detectnet_v2_2.ipynbを選択



TAO Tool kit
3.事例：物体認識

① detectnet_v2：物体認識

- 概要：車、バン、トラック、歩行者、サイクリストのデータからresnet18の学習モデルを使って、検出対象を3個（車、サイクリスト、歩行者）に絞り、学習モデルを作成。その後、prune, 量子化などを使い、最終的にtensorRT用のモデルとして出力。Taoで最初に使ったモデルに比べて、12倍に高速化し、正解率もほぼ同じを実現。
- https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/detectnet_v2.html
- detectnet_v2_2.ipynbを選択
- 認識率と推定時間の各モデル別推移

Tao当初　評価結果	tao平均推論時間(sec)：①	prune後の再学習結果	量子化後の評価結果	tensorRT後の評価結果	tensorRT後の平均推論時間(sec)：②	高速化倍率：①/②	tao学習、prune後再学習時間の合計(hour)
class name　average precision (in %) ----------　---------- car　79.5795 cyclist　80.3184 pedestrian　65.9508	0.005441	class name　average precision (in %) ----------　---------- car　80.2562 cyclist　81.5315 pedestrian　65.9992	class name　average precision (in %) ----------　---------- car　75.3572 cyclist　77.1818 pedestrian　65.6729	class name　average precision (in %) ----------　---------- car　74.9865 cyclist　77.9812 pedestrian　66.0452	0.000436	12.48	4

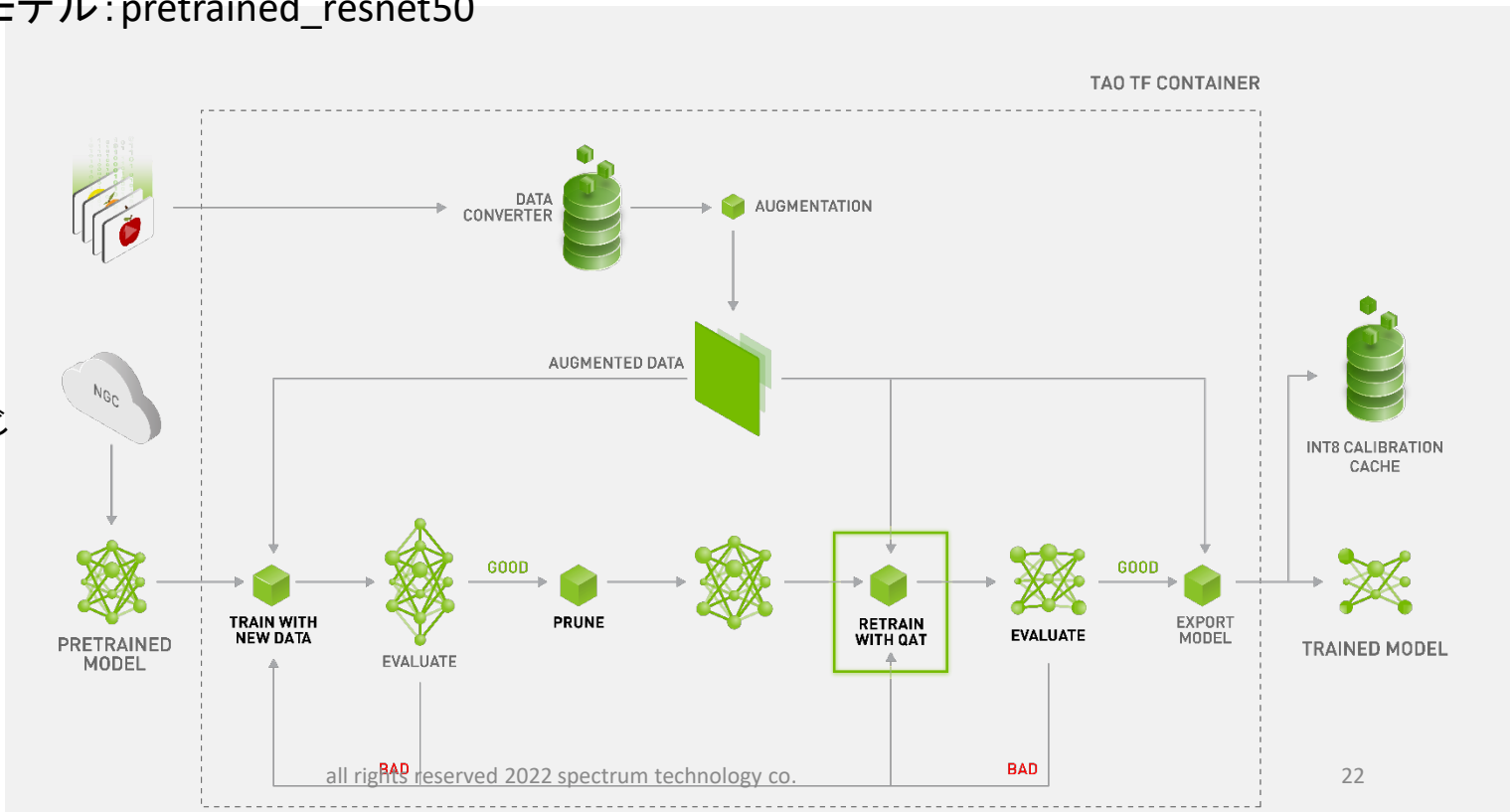
最初の推論より、12倍高速化

TAO Tool kit
4.事例：セグメンテーション

⑪ unet：セグメンテーション

- 概要：医療系の画像を使って、血管などをセグメンテーションする。Unet_isbiを参照
- https://docs.nvidia.com/tao/tao-toolkit/text/semantic_segmentation/unet.html
- データセット： isbi
- 事前学習モデル：pretrained_resnet50

全体図
Detectnetと同じ



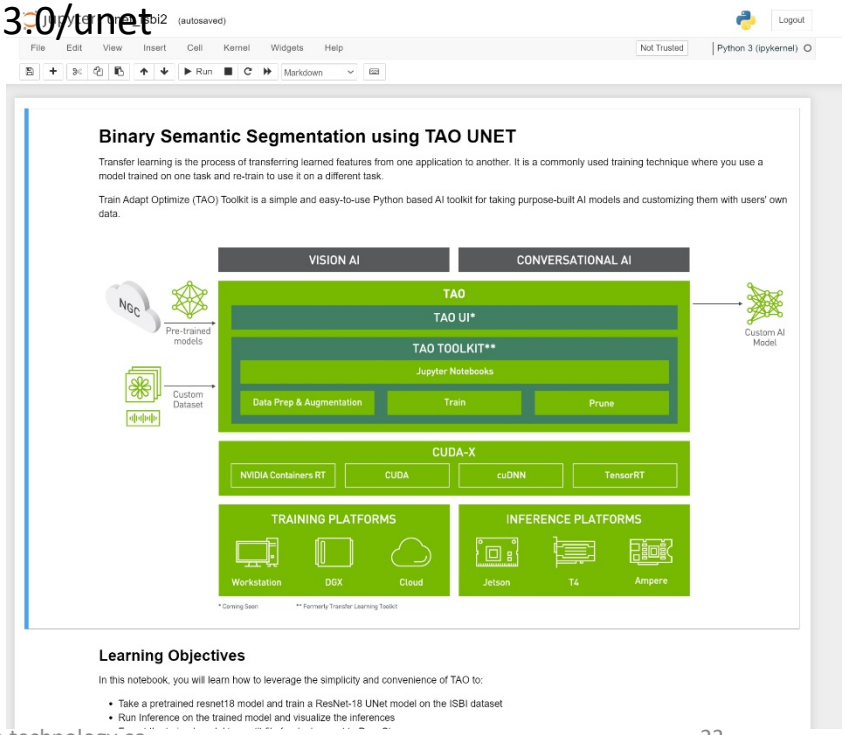
入力コマンド
$ cd
/home/masa/Documents/tao/cv_samples_v1.3.0/unet
$ jupyter notebook

TAO Tool kit
4.事例：セグメンテーション

⑪ unet：セグメンテーション

- 概要：医療系の画像を使って、血管などをセグメンテーションする。Unet_isbiを参照
- https://docs.nvidia.com/tao/tao-toolkit/text/semantic_segmentation/unet.html
- データセット： isbi
- 事前学習モデル：pretrained_resnet50

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/unet
(venv_py36) masa@ubuntu2 :$ jupyter notebook
unet_isbi2.ipynbを選択
Jupyterの各処理は省略



Spectrum Technology

入力コマンド
$ cd
/home/masa/Documents/tao/cv_samples_v1.3.0/unet
$ jupyter notebook

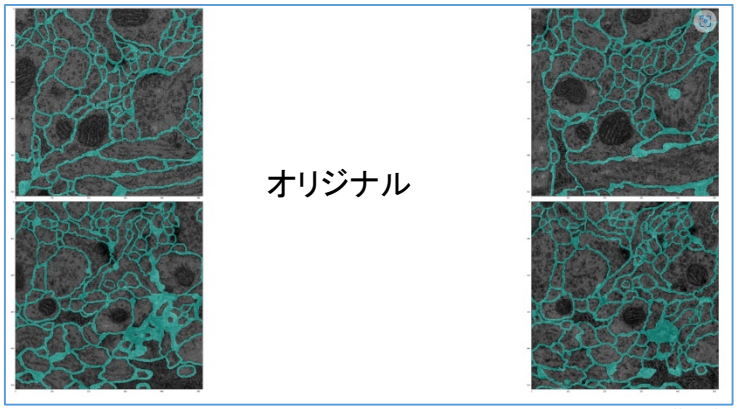
TAO Tool kit
4.事例：セグメンテーション

⑪ unet：セグメンテーション

- 概要：医療系の画像を使って、血管などをセグメンテーションする。Unet_isbiを参照
- https://docs.nvidia.com/tao/tao-toolkit/text/semantic_segmentation/unet.html

unet_isbi2.ipynbを選択

tao当初評価結果	tao平均推論時間（sec）：①	prune後の再学習結果	量子化後の評価結果	tensorRT後の評価結果	tensorRT後の平均推論時間(sec):②	高速化倍率：①/②	tao学習、prune後再学習時間の合計(hour)
Recall : 0.8171335756778717 Precision: 0.831825852394104 F1 score: 0.8241343816116111 Mean IOU: 0.7151066660881042	Throughput Avg: 75.287 img/s Latency Avg: 48.422 ms Latency 90%: 59.762 ms Latency 95%: 61.934 ms Latency 99%: 66.18 ms	Recall : 0.8249054551124573 Precision: 0.8128768801689148 F1 score: 0.8186391778106556 Mean IOU: 0.7068162858486176			10/10 [00:00<00:00, 77961.04it/s]	1036.03	0.333333



Spectrum Technology

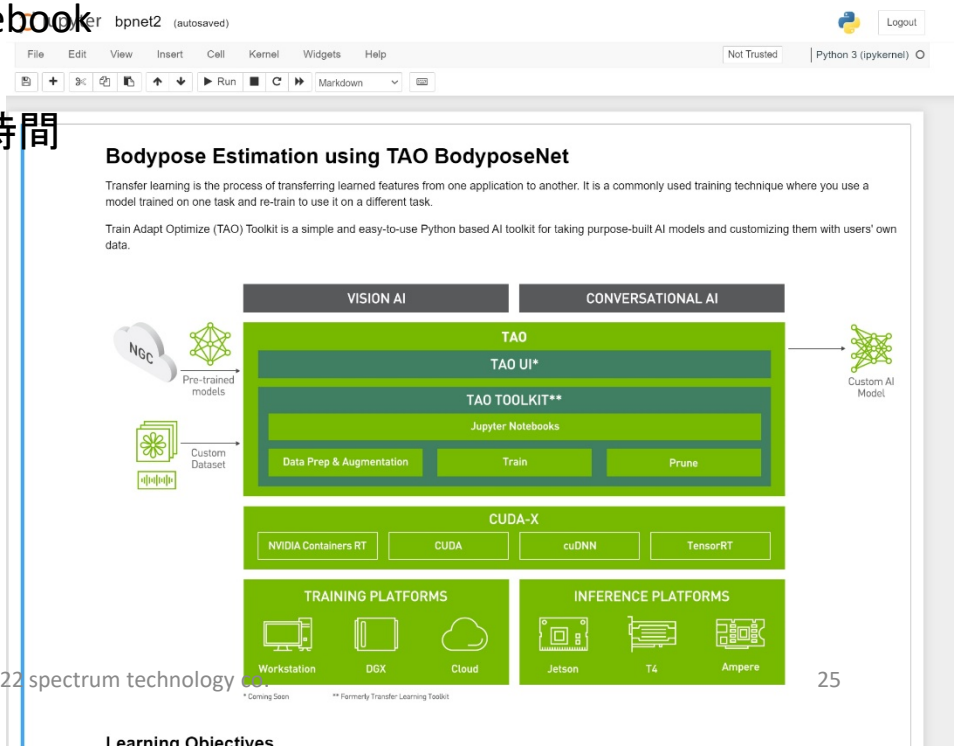
入力コマンド
$ cd
/home/masa/Documents/tao/cv_samples_v1.3.0/bpnet
$ jupyter notebook

TAO Tool kit
5.個別事例

⑬ BodyPoseNet

- 概要：body pose netは、からだの目、耳、手首などの位置を検出します。
- https://docs.nvidia.com/tao/tao-toolkit/text/bodypose_estimation/bodyposenet.html
- データセット： coco
- 事前学習モデル：bodyposenet_vtrainable_v1-2.0

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/bpnet
(venv_py36) masa@ubuntu2 :$ jupyter notebook
bpnet2.ipynbを選択
Jupyterの各処理は省略,学習時間は、28時間



Spectrum Technology

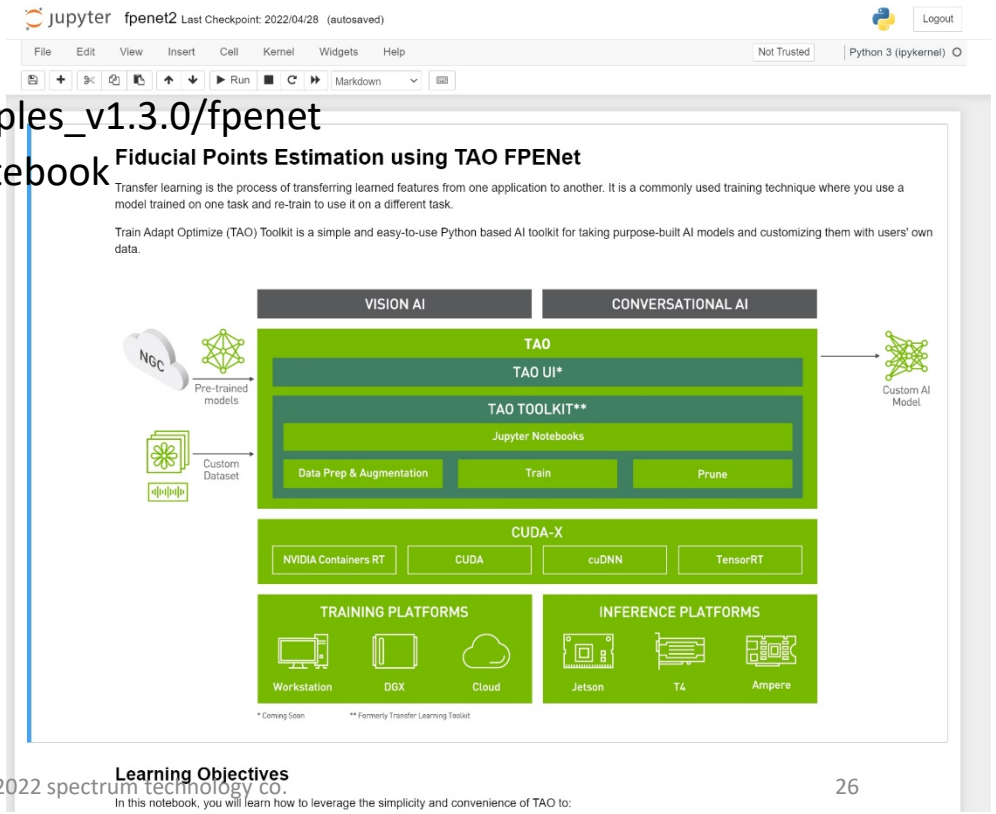
TAO Tool kit
5.個別事例

入力コマンド
$ cd
/home/masa/Documents/tao/cv_samples_v1.3.0/fpenet
$ jupyter notebook

⑰ Fpenet

- 概要：顔の部位(Eyes,Nose,Mouth,Eyebrows,Chin,HP,Pupil,Ears,)を検出します。
- https://docs.nvidia.com/tao/tao-toolkit/text/facial_landmarks_estimation/facial_landmarks_estimation.html
- データセット： AFW
- 事前学習モデル：fpenet:trainable_v1.0

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/fpenet
(venv_py36) masa@ubuntu2 :$ jupyter notebook
fpenet2.ipynbを選択
Jupyterの各処理は省略



Spectrum Technology

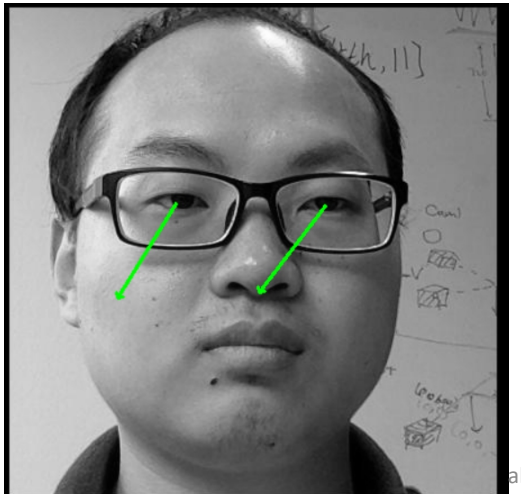
入力コマンド
$ cd
/home/masa/Documents/tao/cv_samples_v1.3.0/gazenet
$ jupyter notebook

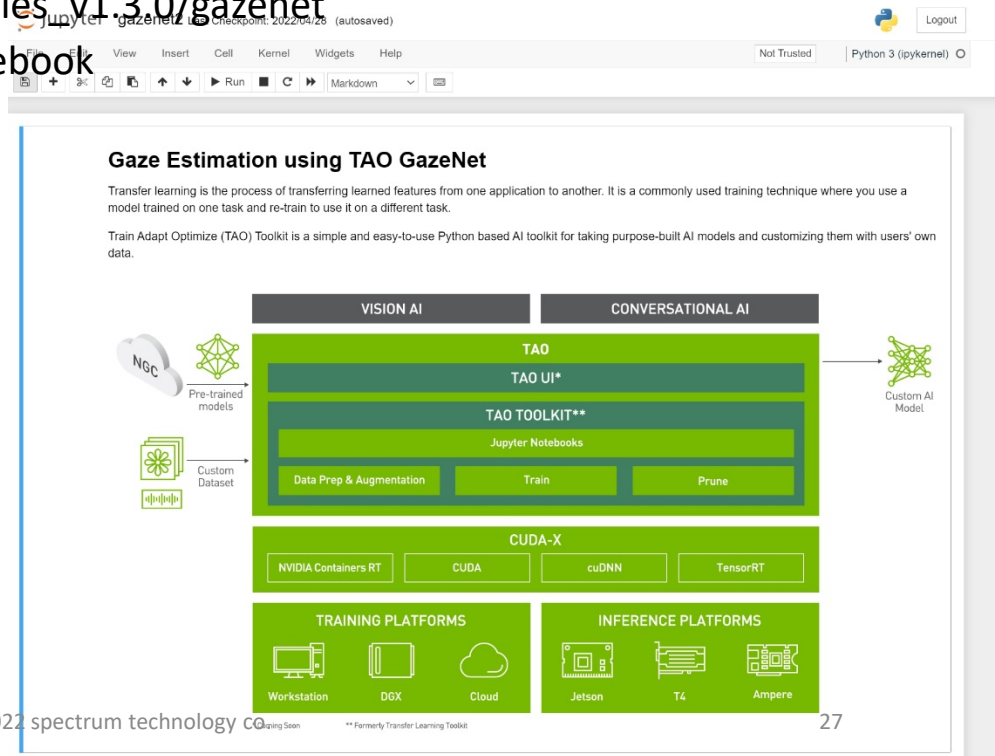
TAO Tool kit
5.個別事例

⑱ gazenet

- 概要：顔から視線を推測。
- https://docs.nvidia.com/tao/tao-toolkit/text/gaze_estimation/gaze_estimation.html
- データセット：　MPIIFaceGaze
- 事前学習モデル：gazenet:trainable_v1.0

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/gazenet
(venv_py36) masa@ubuntu2 :$ jupyter notebook
gazenet2.ipynbを選択
Jupyterの各処理は省略





Spectrum Technology

TAO Tool kit
5.個別事例

入力コマンド
$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/lprnet
$ jupyter notebook

21　lprnet

- 概要：ナンバープレートの認識。
- https://docs.nvidia.com/tao/tao-toolkit/text/character_recognition/lprnet.html
- データセット：　OpenALPR benchmark
- 事前学習モデル：pretrained_lprnet_baseline18

US版

$ cd /home/masa/Documents/tao/cv_samples_v1.3.0/lprnet
(venv_py36) masa@ubuntu2 :$ jupyter notebook
lprnet2.ipynbを選択
Jupyterの各処理は省略

読み取り結果
wts-lg-000076.jpg:4GLS168