

V1.0 2022/6/1

## 抜粋版

# はじめての映像AI開発キット

### nvidia deepstreamを使って映像AI開発の体験と開発期間短縮 学習編(deepstream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



- ubuntuについて
- Linux基本コマンド
- ubuntu基本操作
- 日常運用(ウイルススキャン、更新)
- 開発キット 概念図
- 開発キット 全体像
- ハード概要
- ソフト概要

2.

- Deepstream SDK ۲
  - 概要 1.
    - (1) 概要 2 利用方法 3 アプリー覧 基本, tao連携 **(4**) python仮想環境 (5) Deepstream 動作方法 事例:基本アプリ(python版) 1 deepstream-test1 2 deepstream-test2 3 deepstream-test3 4 deepstream-test1-usbcam (5) deepstream-test1-rtsp-out 6 deepstream-imagedata-multistream  $\overline{\mathcal{O}}$ deepstream-ssd-parser 8 deepstream-optical flow 9 deepstream-segmentation
    - (10)deepstream-nvdsanalytics (11) runtime\_source\_add\_delete (12) deepstream-imagedata-multistream-redaction (13)

11

12

13

13

<u>14</u>

16

16

<u>17</u>

18

<u>19</u>

<u>21</u>

<u>22</u>

<u>23</u>

24

<u>25</u>

<u>26</u>

27

28

29

抜粋版のため目次と異な ります

deepstream-rtsp-in-rtsp-out all rights reserved 2022 spectrum technol 30 co.



• [	Deepstream SDK
-----	----------------

3.

事例	:基本アプリ(c++版)	
1	deepstream-test1	<u>31</u>
2	deepstream-test2	<u>32</u>
3	deepstream-app:カメラ	<u>33</u>
4	deepstream-app:ビデオ	<u>34</u>
5	deepstream-ssd-parser	<u>35</u>
6	objectDetector_FasterRCNN	<u>36</u>
$\overline{\mathcal{O}}$	objectDetector_Yolo	<u>37</u>
8	deepstream-dewarper-test	<u>38</u>
9	deepstream-opticalflow	<u>39</u>
10	deepstream-user-metadata-test	<u>40</u>
1	deepstream-image-decode-test	<u>41</u>
(12)	deepstream-segmentation	<u>42</u>
(13)	deepstream-gst-metadata-test	<u>43</u>
14)	deepstream-infer-tensor-meta-app	<u>44</u>
(15)	deepstream-perf-demo	<u>45</u>
16	deepstream-preprocess-test	<u>46</u>
1	deepstream-3d-action-recognition	<u>47</u>
18	deepstream-nvdsanalytics	<u>48</u>
(19)	deepstream-opencv-test	<u>49</u>
20	deepstream-image-meta-test	<u>50</u>
21	deepstream-appsrc-test	<u>51</u>
22	deepstream-transfer-learning-app	<u>52</u>
23	deepstream-mrcnn-test	<u>53</u>

Deepstream SDK

4.	事例:	:TAO連携	
	1	TrafficCamNet	<u>54</u>
	2	PeopleNet-ResNet34	<u>55</u>
	3	Dashcamnet	<u>56</u>
	4	FaceDetectIR	<u>57</u>
	5	VehicleMakeNet	<u>58</u>
	6	vehicletypenet	<u>59</u>
	$\overline{\mathcal{O}}$	License plate detection	<u>60</u>
	8	License plate recognition net	<u>61</u>
5.	事例:	:TAO応用	
	1	2D Body Pose Estimation	<u>62</u>
	2	Facial Landmark	<u>63</u>
	3	emotionnet	<u>64</u>
	4	Gazenet	<u>65</u>
	5	Gesturenet	<u>66</u>
	6	MaskRCNN	<u>67</u>
	$\bigcirc$	DetectNet_V2	<u>68</u>
	8	image_classification	<u>69</u>
	9	PeopleSegNet	<u>70</u>
	10	People Semantic Segmentation	<u>71</u>
6.	Grap	oh composer	
	1	アプリ ワークフローdeepstream-test1	<u>72</u>
	2	アプリ作成方法:deepstream-test1 その2	<u>78</u>
	3	deepstream-test2	<u>80</u>
	4	deepstream-test3	<u>81</u>
	(5)	deepstream-runtime-src-add-del	<u>82</u>
	6	deepstream-app	<u>83</u>
	$\overline{\mathcal{O}}$	deepstream-audio	<u>84</u>
	8	deepstream-camera	<u>85</u>



## Ubuntu運用マニュアル

# ubuntu®

- 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

## ubuntu®

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 てのコマンド共通(マイナスを2個と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 ③ ユーザ権限、プロセス他 スーパーユーザ(root)に切り替え、パスワードを入力 masa@ubuntu :~ \$ su -ルート権限で各種コマンドを実施します。 masa@Ubuntu:~\$ sudo 現状の動いているプロセスを表示 masa@ubuntu:~\$ ps a 特定のプロセスを強制終了 masa@ubuntu:~\$ kill パッケージのインストールなどに使用 masa@ubuntu:~\$ apt-get install pkg 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。 モジュール、usb、メモリ、HDDなどの表示 linuxのモジュールリスト表示 masa@ubuntu:~\$ lsmod 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®

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: Document
s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

# ubuntu®

- 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。

masa@ubuntu: ~	
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)	パターンファイル更新
<pre>masa@ubuntu:~\$ sudo freshclam Fri Jul 12 09:58:58 2019 -&gt; ClamAV update process started 2019 Fri Jul 12 09:58:58 2019 -&gt; ^Your ClamAV installation is C Fri Jul 12 09:58:58 2019 -&gt; ^Local version: 0.100.3 Recomm Fri Jul 12 09:58:58 2019 -&gt; DON'T PANIC! Read https://www. pgrading-clamav Fri Jul 12 09:58:58 2019 -&gt; main.cvd is up to date (versio f-level: 60, builder: sigmgr) Fri Jul 12 09:58:58 2019 -&gt; daily.cld is up to date (versio 19, f-level: 63, builder: raynman) Fri Jul 12 09:58:58 2019 -&gt; bytecode.cvd is up to date (versio f-level: 63, builder: neo) masa@ubuntu:~\$ sudo clamscaninfectedremoverecursio</pre>	\$ 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 clamscaninfectedremoverecursive 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。

	masa@ubuntu: ~	
ファイル(F) 編集(E) 表示(V) あ	検索(S) 端末(T) ヘルプ(H)	
^C		
masa@ubuntu:~\$ sudo apt-g	et update	
ヒット:1 http://jp.orchev	e.abanta.com/ubuntu bionic InRelease	
取得:2 http://jp.archive.	ubuntu.com/ubuntu bionic-updates InRelea	ase [88.7 kB]
無視:3 http://dl.google.	masa@ubuntu	
取得:4 http://jp.archive	masa@obonco	
無視:5 https://developer	ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)	
tu1804/x86_64 InRelease	N: ターゲット CNF (main/cnf/Commands-all) は	/etc/apt/sources.list.d/google-chro
無視:6 http://developer.m	ne.list:3 と /etc/apt/sources.list.d/google.li	ist:1 で複数回設定されています
InRelease	nasa@ubuntu:~\$ sudo apt-get upgrade	
ヒット:7 https://develop	パッケージリストを読み込んています完了	
untu1804/x86_64 Release	衣仔関係ツリーを作成しています 出能は起た詰み取っています。 ウマ	
ヒット:8 http://develope1	K態情報を読み取つています元」 マップグレードパッケージを捨出しています 一点	27
64 Release	リックシンレードバックーンを検出していより・・・ ガ い下のパッケージは保密されます・	5 J
ヒット:9 http://archive. ^	libol1-mesa-dri libxatracker2 mesa-va-drive	rs mesa-vdpau-drivers
ヒット:10 http://archive」	以下のパッケージはアップグレードされます:	
ヒット:11 http://dl.goog	firefox firefox-locale-en firefox-locale-ja	gnome-settings-daemon
取得:13 http://security.	gnome-settings-daemon-schemas libsysmetrics:	1 ubuntu-report
取得:16 http://security.	アップクレード:7個、新規インストール:0個、	削除: 0 個、保留: 4 個。
9 kB] 5	14.4 MB のアーカイノを取得する必要かめります。 この操作後に追加で すっすっしゃ のディスク容量が逃	当時されます
取得:17 http://security.	この採TF夜に迫加で 4,019 KB のナイスク谷里が汚 続行  キオか2 [v/b] v	月夏ごれより。
50 kB]	収得:1 http://ip.archive.ubuntu.com/ubuntu bi	onic-undates/main_amd64_gnome-setti
取得:18 http://security.	ngs-daemon-schemas all 3.28.1-Oubuntu1.3 [12.9	9 kB1
56 kB]	取得:2 http://jp.archive.ubuntu.com/ubuntu bio	onic-updates/main amd64 gnome-setti
n	ngs-daemon amd64 3.28.1-0ubuntu1.3 [316 kB]	
I	収得:3 http://jp.archive.ubuntu.com/ubuntu bio	onic-updates/main amd64 libsysmetri
C	s1 amd64 1.3.2 [1,475 kB] 现但 4 bbbs (/accurate bbs bbs bbs bbs bbs bbs bbs bbs bbs bb	in an antitudantin and the first surder
E	X177:4 NLLP://Security.ubuntu.com/ubuntu bion1 1 68 0+build3-0ubuntu0 18 04 1 [49_8 MB]	ic-security/main amd64 firefox amd6
	all rights reserve	ad 2022 spectrum technology co
		a zozz spectrum teennology co.

# ubuntu®

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

#### 



### ST Spectrum Technology 開発キット 全体像(deepstream版)





#### 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(deepstream版)		32GB USB3.0 deepstream 関連ソフト		
お客様準備品				
PC本体	сри	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		

Jetsonにも搭載可能で すが、今回は、Ubuntu x86\_64で紹介 ST Spectrum Technology 2.ソフトウエア概要 ①ソフトウェア一覧

#### 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
プログラム言語	python3	3.6.9	
deepstream	deepstream	6.0.1	
	Graph composer	1.0.0	ローコード・ツール
ドライバ	Nvidia-driver	470	
AI用プログラム	tensorRT	8.0.1	
映像	Gstreamer	1.14.5	
各種モジュール	Jupyter notebook、 matplotlibなど多数のpipラ イブラリ		

## Deepstream SDK

- 1. 概要
- ① 概要
  - 開発者は、DeepStreamを使用して、AIベースのビデオ、 オーディオ、および画像分析のためのシームレスなスト リーミングパイプラインを構築できます。DeepStreamは、 開発者にC/C++、Pythonで開発するオプションを提供する ことで開発の柔軟性をもたらします。DeepStreamには、さ まざまなハードウェアアクセラレーションプラグインと拡張 機能が付属しています。

② 利用方法

- DeepStreamは、開発者と企業の両方向けに構築されており、最先端のSSD、YOLO、FasterRCNN、MaskRCNNなどの一般的なオブジェクト検出およびセグメンテーションモデルに対して広範なAIモデルのサポートを提供します。カスタム関数とライブラリをDeepStreamに統合することもできます。
- Deepstreamは、ラピッドプロトタイ ベルのソリ 9 バスを選択 るこ きに đ もで ح d NVIDIA 統合 Serverとのネ こPvTorchや とのオ TensorFlowな プロイできます およびバッ ・GPU、マノ マノ の推論にNVIDIA TensorRTを使用すると、可能な ンパフォーマンスを実現できます。 チ処理のサポ 限り最高のパフォ
- <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Overview.html</u>

公式サイト <u>https://developer.nvidia.com/ja-jp/deepstream-sdk</u>

#### DEEPSTREAM SDK



#### ST Spectrum Technology Deepstream SDK 1. 概要

#### ③. アプリー覧 その1 基本

id	区分	アプリ名	概要	deepstream版 python 番号は、目次	deepstream版 c++ 番号は目次	使用モデル
	1物体認識	deepstream-test1	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施	●1	•1	resnet10.caffemodel
	2物体認識	deepstream-test2	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvtracker → nvinfer (secondary classifier) → nvdsosd → rendererの順で実施。2 段階識別	•2	●2	resnet 10. caffemodel
	3物体認識	deepstream-test3	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。複数ファイルを同時処理	●3		resnet 10. caffemodel
	4usbcam	deepstream-test1-usbcam	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。Test1のusbcameraのバージョン	●4		resnet 10. caffemodel
	5rtsp	deepstream-test1-rtsp-out	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。Test1のrtspのバージョン	●5		resnet 10. caffemodel
	6rtsp	deepstream-imagedata-multistream	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。Test3の複数画面のバージョン	●6		resnet 10. caffemodel
	7物体認識	deepstream-ssd-parser	物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。tritonサーバを使って推論	▲7エラー	●5	ssd model
	8光の流れ	deepstream-opticalflow	nvofvisual gstreamer pluginを使って、光の流れに変換	●8	●9	nvofvisual gstreamer plugin
	9 <sup>セグメンテー</sup> ション	deepstream-segmentation	industrial,semanticの2つパターンを使って、物体をセグメンテーション	●9	●12	nvinfer
1	.0動線	deepstream-nvdsanalytics	動きの方向を分析	●10	●18	resnet10.caffemodel
1	1動画編集	runtime_source_add_delete	nvstreammux -> nvinfer -> nvtracker -> nvtiler -> nvvideoconvert-> nvdsosd -> displaysink、 動画を加えたり、削除したりして表示	●11		nvstreammux
1	2物体認識	deepstream-imagedata- multistream-redaction	mp4の複数をh264変換しストリーム、顔を検出して、ぼかしを入れる	●12		
1	.3rtsp	deepstream-rtsp-in-rtsp-out	ファイルをストリームして、vlcでモニタする	●13		nvinfer
1	4物体認識	deepstream-app	汎用モデル、カメラ、ビデオによる物体認識		●3,4	resnet_int8
1	5物体認識	objectDetector_FasterRCNN	物体識別は、FasterRCNNによる識別		●6	VGG16_faster_rcnn_fi nal.caffemodel
1	.6物体認識	objectDetector_Yolo	物体識別は、Yolov3, yolov3_tinyによる識別		●7	yolo
1	.7物体認識	deepstream-dewarper-test	物体識別は、360度カメラ対応、csv出力による識別		●8	不明
1	8メタデータ	deepstream-user-metadata-test	16byteのメタデータを出力		●10	resnet10.caffemodel
1	.9物体認識	deepstream-image-decode-test	mjpegの物体認識		●11	resnet10.caffemodel
2	20メタデータ	deepstream-gst-metadata-test	Gst-nvstreammux pluginを使って、メタデータを出力		●13	resnet10.caffemodel
2	1メタデータ	deepstream-infer-tensor-meta-app	nvinfer tensor を、メタデータとして出力		●14	resnet10.caffemodel
2	2物体認識	deepstream-perf-demo	single channel cascadeと二次認識のデモ		▲15エラー	resnet10.caffemodel
2	3物体認識	deepstream-preprocess-test	画面を分割して試験		●16	resnet10.caffemodel
2	4物体認識	deepstream-3d-action-recognition	3d,2dでの物体認識		▲17エラー	resnet18
2	25物体認識	deepstream-opencv-test	opencvを使った物体認識、認識するとぼかしが入る		●19	resnet10.caffemodel
2	6メタデータ	deepstream-image-meta-test	meta dataを検出し、検出部分をjpegで保存		●20	resnet10.caffemodel
2	27物体認識	deepstream-appsrc-test	AppSrc and AppSink 利用し、i420で出力		●21	resnet10.caffemodel
2	8物体認識	deepstream-transfer-learning-app	転移学習を行い、正解率の低いものを抽出		•22	resnet10.caffemodel
2	29 29 ション	deepstream-mrcnn-test	mask rcnnを使用したセグメンテーション		▲23未実施	resnet 10. caffemodel

#### ST Spectrum Technology Deepstream SDK 1. 概要

#### ③. アプリー覧 その2 tao連携

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 deepstream版のエ
4	YOLOV3	darknetで有名なyolo、高速認識が売り	•	●3 7 ラーは、 engine生
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)	•	converterのエラー
7	RetinaNet	ー般的には、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	•	个明
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	レヨン 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 <b>エラ</b> 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	顏識別		•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	人体おグ後代すreselled 2022 spectrum technology co.		•5_10 15
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
	<ul> <li>区分</li> <li>1物体検出</li> <li>2</li> <li>3</li> <li>4</li> <li>5</li> <li>6</li> <li>7</li> <li>8</li> <li>9</li> <li>10セグメンテージ</li> <li>11</li> <li>12個別事例</li> <li>13</li> <li>14</li> <li>15</li> <li>16</li> <li>17</li> <li>18</li> <li>19</li> <li>20</li> <li>21</li> <li>22</li> <li>23</li> <li>24</li> <li>25</li> <li>26</li> <li>27</li> <li>28</li> <li>29</li> <li>30</li> <li>31</li> <li>32</li> <li>33</li> <li>34</li> </ul>	区分アプリ名1<物体検出	区分         アブリ名         概要           1<	区分         アブリ名         機要         Ionix           1         物体検出         DetectNet_V2         単、パン、トラック、歩行者、サイクリスト、歩行者1に対し、空音モデルを 作成、その後、prune_見ておなどをしい、最終的にencontPotチルとし て出力、Taoで見知に使ったモデルに比べて、12倍に高速化し、正解平もは に同とを実現         ●           2         FasterRCNN         開上のデータセット、事前学習モデルで、FasterRCNNモデルで学習する。正 に同と実現         ●           3         SSD         トワークで発出する。一般には、認識構成生高く高速といわれているが?結 果は、良力。         ●           4         YOLOV3         darknetで得るなyolo。高速認識が売り、新たな開発者が担当 darknetで得るなyolo。高速認識が売り、新たな開発者が担当 のはたまで見たり、新たな開発者が担当 darknetで得るなyolo。高速認識が売り、新たな開発者が担当、小型モデル 版(程い)         ●           5         YOLOV3         darknetで得るなyolo。高速認識が売り、新たな開発者が担当 darknetで得るなyolo。高速認識が売り、新たな開発者が担当、小型モデル 版(程い)         ●           7         RetinaNet         ・・一般的には、没層間で執いしていたものを、速度を補持したますで、構成が高 い一般機械出デルができないかと考え、RetinaNetが発きまれたした         ●           8         efficientdet         efficientdet cefficient detを使うた物体認識         ●           9         DSSD         deconvolutional single stot detector(ISD)         ●           10         UNET         Cageのを行いたきす         ●           12         UNET         Cageのがの機能と行いたちの目、頂、手音などの位置を使出します。         ●           13         Depret         Dody pose net14、から行び目、耳、手音などの位置を使出します。         ●         ●           13         Depret         動から積税設置、クーへ、suxban         ●         ●

### Deepstream SDK

1. 概要

#### 事前準備と動作方法

- ④ python仮想環境
  - 仮想環境は使用しません。
- ⑤ Deepstream SDKの動作方法
  - •動作方法には以下の二つがあります。
  - A) Python
    - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Python\_Sample\_Apps.html</u>
    - python3で動作させます。以下例です。

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/deepstream\_python\_apps/apps/deepstream-test1

\$ python3 deepstream\_test\_1.py tokyo\_keiba1.h264

- B) C++
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_C\_Sample\_Apps.html</u>

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-test1

\$ deepstream-test1-app tokyo\_keiba1.h264

入力コマンド \$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/deepstream\_python\_apps/apps/deepstre am-test1 \$ python3 deepstream\_test\_1.py tokyo\_keiba1.h264

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-test1 \$ deepstream-test1-app tokyo\_keiba1.h264

> 他にgraph composerがノ ンコード用としてあります が、省略します。

> > エラー発生の場合は、同フォルダ内 のdstest1\_pipe\_config.txtのモデル などの位置を要チェック

### ST Spectrum Technology Deepstream SDK 2.事例:基本アプリ(python版)

- ① deepstream-test1:ビデオ物体認識
  - 概要:物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Python\_Sample\_Apps.html</u>
  - 識別対象: Vehicle, RoadSign, TwoWheeler, Person
  - 使用モデル:resnet10.caffemodel

\$ cd /opt/nvidia/deepstream/deepstream-

6.0/sources/deepstream\_python\_apps/apps/deepstream-test1

\$ python3 deepstream\_test\_1.py tokyo\_keiba1.h264



#### 入力コマンド ST Spectrum Technology Deepstream SDK 2.事例:基本アプリ(python版)

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/deepstream python apps/apps/deepstream-segmentation \$ python3 deepstream segmentation.py dstest segmentation config industrial.txt dog.jpg output \$ python3 deepstream\_segmentation.py

- 9 deepstream-segmentation:写真物体認識
  - 概要:industrial,semanticの2つパターンを使って、物体をセグメンテーション
  - https://docs.nvidia.com/metropolis/deepstream/6.0.1/devguide/text/DS Python Sample Apps.html
  - 使用モデル: nvinfer

\$ cd /opt/nvidia/deepstream/deepstream-

6.0/sources/deepstream python apps/apps/deepstream-segmentation

\$ python3 deepstream\_segmentation.py dstest\_segmentation\_config\_industrial.txt dog.jpg output

\$ python3 deepstream segmentation.py dstest segmentation config semantic.txt dog.jpg











入力コマンド \$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-test1 \$ deepstream-test1-app tokyo\_keiba1.h264

- ① deepstream-test1:ビデオ物体認識
  - 概要:物体識別は、single H.264 stream: filesrc → decode → nvstreammux → nvinfer (primary detector) → nvdsosd → rendererの順で実施。
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_C\_Sample\_Apps.html</u>
  - 識別対象: Vehicle, RoadSign, TwoWheeler, Person
  - 使用モデル:resnet10.caffemodel
  - \$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-test1

\$ deepstream-test1-app tokyo\_keiba1.h264





- ⑦ objectDetector\_Yolo:ビデオ物体認識
  - 概要:物体識別は、Yolov3, yolov3\_tinyによる識別
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_ref\_app\_deepstream.html</u>
  - ・識別対象:80分類(人、車など)
  - 使用モデル: yolo
  - \$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/objectDetector\_Yolo
  - \$ deepstream-app -c deepstream\_app\_config\_yoloV3.txt
  - \$ deepstream-app -c deepstream\_app\_config\_yoloV3\_tiny.txt



入力コマンド \$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/objectDetector\_Yolo \$ deepstream-app -c deepstream\_app\_config\_yoloV3.txt \$ deepstream-app -c deepstream\_app\_config\_yoloV3\_tiny.txt

入力コマンド

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstreamnvdsanalytics-test

\$ deepstream-nvdsanalytics-test

file:///opt/nvidia/deepstream/deepstream-

6.0/samples/streams/sample\_720p.mp4

18 deepstream-nvdsanalytics-test:ビデオ物体認識

- 概要:動態の進行方向を表示するの物体認識
- <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_ref\_app\_deepstream.html</u>
- ・識別対象: Vehicle , RoadSign, TwoWheeler, Person,車の場合は、色、メーカ、車種を二次認識
- 使用モデル: resnet10.caffemodel

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstreamnvdsanalytics-test

\$ deepstream-nvdsanalytics-test file:///opt/nvidia/deepstream/deepstream-6.0/samples/streams/sample\_720p.mp4





入力コマンド

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-imagemeta-test \$ deepstream-image-meta-test

- file:///opt/nvidia/deepstream/deepstream-
- 6.0/samples/streams/sample 720p.mp4

② deepstream-image-meta-test:ビデオ物体認識

- 概要:meta dataを検出し、検出部分をjpegで保存
- <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_ref\_app\_deepstream.html</u>
- 識別対象: Vehicle, RoadSign, TwoWheeler, Person,
- 使用モデル: resnet10.caffemodel

事前に、makefileのcuda ver=11.4としsudo make

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/deepstream-imagemeta-test

\$ deepstream-image-meta-test file:///opt/nvidia/deepstream/deepstream-6.0/samples/streams/sample 720p.mp4



#### Deepstream SDK 4.**事例**:TAO連携

- ① trafficcamnet:カメラ物体認識
  - 概要:Webカメラで車、人などを識別
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_TAO\_integration.html</u>
  - 識別対象:トラック、クルマ、人など4種類
  - 使用モデル: DetectNet\_v2-ResNet18

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/trafficcamnet-on-deepstream

\$ deepstream-app -c deepstream\_app\_source1\_trafficcamnet.txt

アクティビ	〒ィ 🔲 不明なプログラム 🔻	6月7日(以)09:55 🔽 🥤 💆 🐂 🗛 🗸 🔥	10) (C) <del>-</del>
9			
• 🔚	☐ \$ DeepStream	Deps/trafficcamnet-on-deepstream	
	Por Contraction	ev-guide/text/DS_TAO_integration.html > ☆ ⇔ □ □ □ ■ ★ □ ● : たいらいになったいのでは、 いっていたいのでは、 いったいのでは、 いっていたいのでは、 いったいのでは、 いったいいいいいいいいいいいいいいいい。	
?		tao使い方メモ20220602-txt 保存(5) 目 ●●● (Droine) ので変更 let ResNet18 6 96% Classifying type of cars as cupe, sedan, truck, etc	
• • -		Aサーバはあがったが、ホスト側とうまくリン DetectNet_v2- ResNet18 1 98% Detect License plates on Vehicles	
	raspberry_ pi_48_bulls 使い方メ	ローカル C webcamで表示 cd /opt/nvidia/deepstrean/deepstrean-6.0/ s/apps/sample_apps/trafficcannet-on-deepstrean CH Nucleonse (CH) CH Nucleonse S/apps/sample_apps/trafficcannet-on-deepstrean	
. 💓	masa@ubuncu2:/opt/nviola/deeps 画能 ファイル(F) 編集(E) 表示(V) 検索( 設定mpl.cpp:313:notifyLoadModelS ubudia/deepstream/deepstream-6. am/config ipfer primary traf	eepstream-subsources/apply/sample_apply/famile_apply/fa	
• 💿	Runtime commands: h: Print this help q: Quit	eoplenet-on- SegNet UNET 3 94.01 Detect people and provide a semantic segmentation mask in an image	
	ar_n ∰ p: Pause 03: r: Resume	pstream-6.6/Sources/ 河、929/) マ [挿入] ned with TAO toolkit are natively integrated for inference with DeepStream. If the model is integrated, it is supported by the reference (deepstream-app.) If the model is not natively integrated in the SDK, you can find a reference	
	**PERF: FPS 0 (Avg) **PERF: 0.00 (0.00) ■K** INFO: <bus_callback:194>:</bus_callback:194>	application on the GitHub repo. See the table below for information on the models supported. For models integrated into despit rease apply, we have provided sample config files for each of the networks. The sample config files are available in the <u>sample/config/ica_pretrained_models</u> [older. The table below also lists config files for each model.	
	開発** INFO: <bus_callback:180>: ubu</bus_callback:180>	the running and the second all rights reserved 2022 spectrum technology co.	
	**PERF: 31.20 (31.11) **PERF: 30.00 (30.51) **PERF: 30.00 (30.33)	Refer README in package Zopt/mvldia/deepstream/deepstream/samples/configs/tao_pretraimed_models/README.md to	

エラーの場合は、 config\_infer\_primary\_traf ficcamnet.txtなどのモデ ルなどの位置を要確認

入力コマンド

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample\_apps/trafficcamnet-ondeepstream \$ deepstream-app -c

deepstream\_app\_source1\_trafficcamnet.txt

++)を使用した応 になります。

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample apps/peoplenet-ondeepstream \$ deepstream-app -c deepstream app source1 peoplenet.txt

エラーの場合は、

config\_infer\_xxx.txtなど

のモデルなどの位置を要 確認

入力コマンド

• 概要:Webカメラを使って人を認識し、ヒートマップ、社会的距離計算に応用

#### https://docs.nvidia.com/metropolis/deepstream/6.0.1/devguide/text/DS TAO integration.html

識別対象: 顔など3種類

ST Spectrum Technology

Deepstream SDK

4.**事例**:TAO連携

② peoplenet:カメラ物体認識

使用モデル: DetectNet v2-ResNet18/34 •

\$ cd /opt/nvidia/deepstream/deepstream-6.0/sources/apps/sample apps/peoplenet-ondeepstream

\$ deepstream-app -c deepstream app source1 peoplenet.txt

7071E	ティ □ 不明なブログラム ◆ DeepStream ● ◎ ●	6月 7日 (2 mple_apps/ 🔏 TAO Toolkit Integration 🗤 🗙 🍕 NVIDIA DeepStream	() 10 : 14 SDK   ×   ႃ႗ \	Yahoo! JAPAN ×	+	V2 c	MN990 A +	. 40 O . □ ×	前出の <u>deepstream</u> app(c++)を使用した「 用編になります。
		← → C ☆ ● docs.nvidia.com/metropolis/deepstream ■ web ■ ビジネス ■ ビジネス 開発 ■ Webメール	m/6.0.1/dev-guid	le/text/DS_TAO_integration.htm 音楽 ■ ランニング ■ コ	nl ゴルフ 🖿 android 🖿	> apple CO	☆ □ 図 题 <b>券 □</b> ) ■ 学校 ■ くらし	2 E	
		B. tao使い方メモ20220602.txt 保存(5) 目 ● ● ◎ ③		TAO toolkit	pretrained models - use ca	ises	, ,,	•	
		deepstream_app_xxxx:fileの位置を変更し ●deepstream-app -c	IName	Network Architecture	Number of classes	Accuracy	Use case		
• >	gnap1 D Documents deepstream_ap	<pre>p_source1_trafficcamnet.txt</pre>	:CamNet	DetectNet_v2- ResNet18	4	83.50%	Detect and track cars		
	publ         Downloads         deepstream           Ja         Music         sources/apps/	cd /opt/nvidia/deepstream/deepstream-6.0/ sample_apps/peoplenet-on-deepstream	eNet	DetectNet_v2- ResNet18/34	3	84%	People counting, heatmap generation, social distancing.		
	masa@ubuntu2: /opt/nvidia/deepstream/deepstrea 面(ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) へル;	m-6.0/sources/apps/sample_apps/peopl 🔵 🗊 🕲 🧕 inferxxxx.txt プ(H)	CamNet	DetectNet_v2- ResNet18	4	80%	Identify objects from a moving object		
. 🗾	設築0:00:18.574074610 8368 0x55d3b768f890 INU ubumpl.cpp:313:notfyLoadModelStatus:eprimary dia/deepstream/deepstream-6.0/sources/appy onfig infer primary peoplemet.txt successfi	FO nvinfer gstnvinfer_i y_gie> [UID 1]: Load new model:/opt/nvi s/sample_apps/peoplenet-on-deepstream/c ulv ashcamnet-on-	etectIR	DetectNet_v2- ResNet18	1	96%	Detect face in a dark environment with IR camera		
•	<pre> Runtime commands:</pre>	cd /opt/nvidia/deepstream/deepstream-6.0/	eMakeNet	ResNet18	20	91%	Classifying car models		
	h: Print this help q: Quit	なしゃ タブ加:8~ (1141行、60列) ▼ [挿入]	leTypeNet	ResNet18	6	96%	Classifying type of cars as coupe, sedan, truck, etc		
	r: Resume	Performance	nse Plate ection	DetectNet_v2- ResNet18	1	98%	Detect License plates on Vehicles		
	■ **PERF: FPS 0 (Avg) 画候**PERF: 0.00 (0.00) 第3 開新	<b>dy</b>	nse Plate ognition	Tuned ResNet18	36(US)/68(CH)	97% (US)/99% (CH)	Recognize characters in License plates. Available in American and Chinese License		24
	ubu** INFO: <bus_callback:180>: Pipeline run</bus_callback:180>	ning DeepStream					plates		
	**PERE: 31.23 (30.97)						Detect and segment		

#### Deepstream SDK 5.**事例**:TAO応用

- ① bodypose2:物体認識
  - 概要:人体の18か所を認識して出力(画像、カメラなど)
  - <u>https://github.com/NVIDIA-AI-</u> <u>IOT/deepstream\_tao\_apps/tree/master/apps/tao\_others/deepstream-bodypose2d-app</u>
  - 識別対象:人体の18か所(鼻、首、肩、膝など)識別
  - 使用モデル: nvstreammux

\$ export LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/opt/nvidia/deepstre am/deepstream/lib/cvcore\_libs

インストール方法は、よ

く確認のこと。

\$ ./deepstream-bodypose2d-app

1 ../../../configs/bodypose2d\_tao/sample\_bodypose2d\_model\_config.txt file:///home/masa/Documents/deepstream\_tao\_apps/apps/tao\_others/deepstreambodypose2d-app/00000000241.jpg ./body2dout





入力コマンド

\$ cd

/home/masa/Documents/deepstream\_tao\_apps/apps/tao\_others/deepstrea m-bodypose2d-app

\$ export

LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/opt/nvidia/deepstream/deepstrea m/lib/cvcore\_libs

\$ ./deepstream-bodypose2d-app

1../../configs/bodypose2d\_tao/sample\_bodypose2d\_model\_config.txt file:///home/masa/Documents/deepstream\_tao\_apps/apps/tao\_others/dee pstream-bodypose2d-app/00000000241.jpg ./body2dout

#### Deepstream SDK 5.**事例**:TAO応用

- ⑥ ds-tao-segmentation:物体認識
  - 概要:動画セグメンテーション(画像、カメラなど)
  - <u>https://github.com/NVIDIA-AI-IOT/deepstream\_tao\_apps</u>
  - 識別対象:セグメンテーション
  - 使用モデル:
  - \$ cd /home/masa/Documents/deepstream\_tao\_apps
  - \$ ./apps/tao\_segmentation/ds-tao-segmentation -c configs/unet\_tao/pgie\_unet\_tao\_config.txt
  - -i /opt/nvidia/deepstream/deepstream-6.0/samples/streams/sample\_720p.h264 -d



入力コマンド

\$ cd /home/masa/Documents/deepstream\_tao\_apps \$ ./apps/tao\_segmentation/ds-tao-segmentation -c configs/unet\_tao/pgie\_unet\_tao\_config.txt -i /opt/nvidia/deepstream/deepstream-6.0/samples/streams/sample\_720p.h264 -d

> 実施前に環境変数を設定するか、bashrcなどへ \$ export LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/opt/nvidia/deepstre am/deepstream/lib/cvcore\_libs

### ST Spectrum Technology Deepstream SDK 6.Graph composer

#### ② アプリ作成方法:deepstream-test1 その2

- グラフ作成方法
- https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_GraphComposer\_Create\_Graph.html
- グラフの起動

\$ sudo composer

- 各コンポーネットの作成
  - 右側のコンポーネント一覧からドラッグアンドドロップします。
  - NvDsSingleSrcInput
    - NvDsSourceExtの配下からdrag and dropします。
    - Prioritiesのところに入力のビデオファイルのパスを入力
    - file:///opt/nvidia/deepstream/deepstream/samples/streams/sample 1080p h264.mp4
  - NvDsStreamMux
    - NvDsMuxDemuxExt配下から
    - batch size: 1,width:1920, height:1080を設定 properties
  - NvDsInferVideo
    - NvDsInferenceExt配下から
    - 使用するモデルを設定
    - NvDsResnet10 4ClassDetectorModelをNvDsSampleModelsExt配下からドラッグ
    - 上記とinfer model configとを接続
  - NvDsOSD
    - NvDsVisualizationExt配下から
  - NvDsVideoRenderer
    - NvDsOutputSinkExt配下から
  - NvDsPerClassObjectCounting
    - NvDsInferenceUtilsExt配下から
  - NvDsScheduler
    - NvDsBaseExt配下から
- 各コンポーネント接続
  - 各コンポーネントのvideo-outとvideo-inをドラッグして接続、右の図のとおり
- 保存
  - File>save asでエラーなしで保存できれば完了 all rights reserved 2022 spectrum technology co.



#### ST Spectrum Technology Deepstream SDK

6.Graph composer

#### ② アプリ作成方法:deepstream-test1 その2

- グラフ作成方法
- <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_GraphComposer\_Create\_Graph.html</u>
- グラフの起動
- \$ sudo composer
- 各コンポーネットの作成
- 各コンポーネント接続
  - 各コンポーネントのvideo-outとvideo-inをドラッグして接続、右の図のとおり
- 保存
  - File>save asでエラーなしで保存できれば完了
- ・ラン
  - deepstream-test1と同じ結果がでます。



入力コマンド \$ sudo composer File:///opt/nvidia/deepstream/deepstream/samples/streams/sample\_1080p \_h264.mp4

#### Deepstream SDK 6.Graph composer

- ④ アプリ事例:deepstream-test3
  - C++と同様の事例
  - 3つのモデル識別:車、人、自転車。画面を4分割 <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Zero\_Coding\_Sample\_Graphs.html</u>
  - \$ sudo composer
  - グラフを開く
  - /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstream-test3/
  - ・ラン
    - コンフィグファイルの設定: platform config file
    - /opt/nvidia/graph-composer/config/target\_x86\_64\_cuda\_11\_4.yaml を選択
    - 環境変数:DISPLAY=:0 を入力
    - ・ラン



入力コマンド \$ sudo composer /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstreamtest3 DISPLAY=:0

### **ST** Spectrum Technology Deepstream SDK 6.Graph composer

- ⑥ アプリ事例:deepstream-app
  - C++と同様の事例
  - 画面を30分割
  - <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Zero\_Coding\_Sample\_Graphs.html</u>
  - \$ sudo composer
  - グラフを開く
  - /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstream-app/
  - source30\_1080p\_dec\_infer-resnet\_tiled\_display\_int8.yaml を選択
  - ・ラン
    - コンフィグファイルの設定: platform config file
    - /opt/nvidia/graph-composer/config/target\_x86\_64\_cuda\_11\_4.yaml を選択
    - 環境変数:DISPLAY=:0 を入力
    - ・ラン



入力コマンド \$ sudo composer /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstream-app DISPLAY=:0

#### Deepstream SDK 6.Graph composer

#### ⑧ アプリ事例:deepstream-camera

- C++と同様の事例
- Webカメラで物体を識別。3分類:車、人、自転車
- <u>https://docs.nvidia.com/metropolis/deepstream/6.0.1/dev-guide/text/DS\_Zero\_Coding\_Sample\_Graphs.html</u>
- \$ sudo composer
- グラフを開く
- /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstream-camera/
- ・ラン
  - コンフィグファイルの設定: platform config file
  - /opt/nvidia/graph-composer/config/target\_x86\_64\_cuda\_11\_4.yaml を選択
  - 環境変数:DISPLAY=:0 を入力
  - ・ラン



入力コマンド \$ sudo composer /opt/nvidia/deepstream/deepstream-6.0/reference\_graphs/deepstreamcamera DISPLAY=:0