

Deepstream SDK

アプリ一覧 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	resnet10.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		resnet10.caffemodel
4	usbcam	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		resnet10.caffemodel
5	rtsp	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		resnet10.caffemodel
6	rtsp	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		resnet10.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	セグメンテーション	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
10	動線	deepstream-nvdsanalytics	動きの方向を分析	●10	●18	resnet10.caffemodel
11	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
12	物体認識	deepstream-imagedata-multistream-redaction	mp4の複数h264変換ストリーム、顔を検出して、ぼかしを入れる	●12		
13	rtsp	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
14	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
15	物体認識	objectDetector_FasterRCNN	物体識別は、FasterRCNNによる識別		●6	VGG16_faster_rcnn_final.caffemodel
16	物体認識	objectDetector_Yolo	物体識別は、Yolov3, yolov3_tinyによる識別		●7	yolo
17	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
18	メタデータ	deepstream-user-metadata-test	16byteのメタデータを出力		●10	resnet10.caffemodel
19	物体認識	deepstream-image-decode-test	mjpegの物体認識		●11	resnet10.caffemodel
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
21	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
22	物体認識	deepstream-perf-demo	single channel cascadeと二次認識のデモ		▲15エラー	resnet10.caffemodel
23	物体認識	deepstream-preprocess-test	画面を分割して試験		●16	resnet10.caffemodel
24	物体認識	deepstream-3d-action-recognition	3d,2dでの物体認識		▲17エラー	resnet18
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
26	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
27	物体認識	deepstream-appsrc-test	AppSrc and AppSink 利用し、i420で出力		●21	resnet10.caffemodel
28	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
29	セグメンテーション	deepstream-mrcnn-test	mask rcnnを使用したセグメンテーション		▲23未実施	resnet10.caffemodel

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1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、メーカー名などを識別		●4_3
24		FaceDetectIR	顔識別		●4_4
25		VehicleMakeNet	車種、メーカーの認識		●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