Jetson Nano による実践 AI

目次

- 1. Jetson Nano の購入
- 2. マイクロ SD カードへのイメージ書き込み(通常)
- 3. 機器のセットアップ
- 4. 最初のブート (電源オン)
- 5. コンピュータを使用するための基礎知識、設定、およびツールの導入
- 6. ネットワークの設定
- 7. カメラのセットアップ
- 8. 推論パッケージ(jetson-inference)のインストール、ビルド
- 9. Jetson Nano オンボード機械学習、推論の実行

1. Jetson Nano の購入

NVIDIA 社から販売されている Jetson Nano Developer Kit には本体のみが含まれてい ます。カメラからの物体検出等のディープラーニング処理を行う場合、別途周辺機器の 購入が必要です。以下、リスト機器の購入をお勧めします。

Jetson Nano Developer Kit

Jetson Nano 本体 https://www.physical-computing.jp/product/1740 ¥13,800 ブート用メインストレージ Micro SD カード (32GB) または Micro SD カード (64GB) ディスプレイモニター 液晶モニター 13.3 インチ HDMI LCD(H) (with case) V2, 1920x1080 Raspberry Pi4 対応 https://www.physical-computing.jp/product/1804 ¥18,900 7インチ HDMI LCD 1024x600 Raspberry Pi 対応 www.amazon.co.jp/gp/product/B07PY5XND3/ref=ppx_yo _dt_b_asin_title_o04_s00?ie=UTF8&psc=1 ¥8,960 HDMI ケーブル 入力デバイス 英語キーボード マウス カメラ MX219-D77 モジュール (Raspberry Pi カメラボード V2) https://www.physical-computing.jp/product/1780 ¥3,980 ファン、固定アクリル板 Jetson Nano 用 ファン&ベースプレートキット https://www.physical-computing.jp/product/1803 ¥1,800

電源 AC アダプター(5V 4.0A) ジャンパーピン

フィジカルコンピューティングラボのネット販売では、本体、マイクロSDカード、電源,ジャンパーピン、HDMIケーブルがセットとなった以下のキットも販売されています。

NVIDIA Jetson Nano コンプリートスターターキット(32GB) https://www.physical-computing.jp/product/1739 ¥ 17,800

NVIDIA Jetson Nano コンプリートスターターキット(64GB) https://www.physical-computing.jp/product/1746 ¥19,800

液晶は 13.3 インチ 1920x1080 でサイズの大きなものとなりますが、キーボード、マウ ス以外、すべて、フィジカルコンピューティングラボで準備できると思います。

ネットワークの接続を Wi-Fi で行いたい場合は、以下をお勧めします。 TP-Link WIFI 無線 LAN 子機 11n/11g/b デュアルモード対応モデル TL-WN725N <u>https://www.amazon.co.jp/gp/product/B008IFXQFU/ref=ppx_yo_dt_b_asin_title</u> <u>_____004_s00?ie=UTF8&psc=1</u> ¥ 790

上記のリストすべて、購入し、動作を確認しています。

後で詳細は述べますが、NVIDIA は、NVIDIA Deep Learning Institute (DLI)という オンライン無料の Deep Learning 学習コースを用意しています。これは登録すること で、誰にでも利用できます。以下のようなコースを無料で受けることができます。 コンピュータビジョン対応のディープラーニングの基礎 Jetson Nano ではじめる AI 入門 DIGITS によるイメージのクラス化 DIGITS による物体の検出 Ten sorRT による Tensor モデルの開発と最適化 もし、この DLI 学習コースを受ける場合、Micro SD カードは 64GB をお奨めします。 通常の SD カードに書き込む Jetpack と DLI 用のイメージとは異なります。ダウンロ ード先も異なります。DLI 用イメージは、通常の Jetpack に PyTorch、JupyterLab 等が 追加され、容量も大きくなっています。 2. マイクロ SD カードへのイメージ書き込み(通常) DLI 学習コース用のイメージ書き込みの詳細は別途記述します。ここでは通常のイメ ージ書き込みを説明します。

マイクロ SD カードは、ブートデバイス、メインストレージとして使用されます。この ため、高速で、十分な容量が求められます。少なくとも 16GB UHS-1 が要求されます。 先の購入リストでリストアップされた S D はこの要求にあっています。

- Jetson Nano Developer Kit 用 SDカードイメージを以下からダウンロードします。 https://developer.nvidia.com/embedded/learn/get-started-jetson-nanodevkit#write-windows
 これは5.2GB程度で数時間かかります。
- SDカードをリードライトできるスロット、またはアダプターをもった、インタ ーネットに接続されたコンピュータを準備します。MAC、LINUX等のコンピ ュータでも可能ですが、ここでは Windows パソコンを使用します。
- SDアソシエーションという規格団体から提供されているSDメモリカード フォーマッターを使用します。
 - 1. 以下から、ダウンロード、インストールします。

https://www.sdcard.org/downloads/formatter/eula_windows/



画面下まで、スクロールし、「Accept」ボタンを押し、ダウンロードします。 ダウンロードフォルダに以下のファイルがダウンロードされます。

SDCardFormatterv5_WinEN.zip

このファイルをクリック、解凍、インストールします。以下SDカードフォーマッターがインストールされます。



- SDカードをPCのスロット、またはアダプターにセットします。
 このとき、PCは特に反応しません。
- 3. SDカードフォーマットターを起動します。

SD Card Fo	rmatter	×
ファイル へ	プ	
カードの選択		
F:¥		•
		更新
カード情報		
種類	SDHC	52
容量	27.84 GB	не
ーフォーマットオブ:	ンョン マット 	
◎ 上書さフォー	יאעי <i>ד</i> י	
CHS7#~~?	ットサイズ調整	
ボリュームラベル		
(
		7ォーマット
	sDロゴ、SDHCロゴ、およびSDXCロゴ()	はSD-3C, LLCの商標です。

SDカードが正常にセットしていれば、カードの選択に表示されます。

- 4. イメージを書き込むSDカードを選択します。
- 5. フォーマットオプションで「クイックフォーマット」を選択します。
- 6. ボリュームラベルは空白のままにします。
- 7.「フォーマット」ボタンを押して。フォーマットを開始します。

3) SDカードへのOSイメージ書き込み

1. Balena 社が提供している無料の書き込みツール balenaEtcher を使用します。 以下のサイトからダウンロード、インストールします。 https://www.balena.io/etcher/

- -× 👴 balenaEtcher - Home × + 🛽 Getting Started With Jetson 🕅 🗙 📪 balenaEtcher - Home → C ☆ ê balena.io/etcher/ ☆ 💿 -時停止中) : 🔢 アプリ 🌨 業績予報 📒 rhino 📒 News 📒 Developper 📒 WebApp 📒 MyProject 📒 MyBlog 📒 MyWork » 📔 その他のブックマーク 😚 balena Etcher \equiv 0 Flash. Flawless. Flash OS images to SD cards & USB drives, safely and easily (+) Select image Download for Windows (x86|x64) v1.5.70 <u>See what's new</u> すべて表示 × Inv-jetson-nano-….zip 、 キャンセルされました

ダウンロードフォルダに以下のファイルがダウンロードされるので、クリックし インストールします。

balenaEtcher-Setup-1.5.70.exe (バージョンは常に更新されています)



2. balenaEtcher を起動します。

🔶 Etcher	a (775) (244	1. mar	den	
				9 -
+			7	
Select image	Multi FlaSB Device			
img, iso, zip, and many more				
😭 balenal	Etcher is an open source project b	🔗 balena		
		-		

- 3. 「Select imaghe」 ボタンで、先に、ダウンロードした Jetson Nano Developer Kit
- 用 SDカードイメージZIPを選択します。 nv-jetson-nano-sd-card-image-r32.3.1.zip 5,423MB

4. SDカードを挿入していないならば、SDスロット、またはアダプターにファ ーマットしたSDカードをセットします。

5.「Flash」ボタンでイメージの書き込みを実行します。書き込み後、Windows が SDカードをリードできないとのエラーメッセージを表示する場合がありますが、 キャンセルします。

6. SDを取り出します。SDカードへのイメージ書き込みは完了です。

3. 機器のセットアップ

購入部材のファン、固定アクリル板、Jetson Nano 用ファン&ベースプレートキット、 及び、Jetson Nano 本体、ジャンパーピンを取り出します。





1) アクリル板の両面の保護シートをはがします。

2) 4 セットのビス、ナット、スペーサーでアクリル板を Jetson Nano 本体の底に取り付けます。アクリル板は、方向性があるので、正しく取り付けます。
3) アクリル板底の4 隅に透明のすべり止めを貼り付けます。



4) ファンを放熱板にとりつけます。放熱板はアルミで、タップを切っていません。ま た板圧も大きくありません。このため、6 角ビスでネジを切りながら締めてゆきますが、 強くすると、ネジがつぶれてしまいます。

5) ファンのコネクタを本体の FAN J15 のヘッダーに挿入します。ファンのコネクタ は3ピンで本体ヘッダーは4ピンですが、キーロックがあるため、誤挿入はないと思い ます。

GND	FAN 線	黒
5V	FAN 線	赤
Tachometer	FAN 線	黄
PWM		

	100	
• •	• •	7
PWM	5V	r i
TA	CH	ĠND

6) 電源ジャンパーピンの設定をします。DC ジャックから電源を供給する場合は、J48 のヘッダーをジャンパーピンでショートします。USBから供給する場合はオープン にします。



7) Jetson Nano モジュールの背面にある SD スロットに、イメージを書き込んだ S D カードを挿入します。



8) 液晶モニターに電源を入れ、HDM I ケーブルで Jetson Nano 本体に接続します。

9) キーボード、マウスを接続します。

10)最後にDC電源をDCジャックに接続します、電源が供給されますと、自動的 にブートが開始されます。



4. 最初のブート(電源オン)

Jetson Nano が電源オンされると、すぐに、マイクロUSBコネクタ横のグリーンLE Dが点灯します。またファインも回転を始めます。

初めてのブート(電源オン)であれば、以下の初期設定が要求されます。

NIVIDIA Jetson Nano ソフトウエア使用許諾

EULA(End-User License Agreement)確認

 \Box I accept the terms of these licenses

にチェックマークをオンし、[continue]ボタンを押し、次に進みます。

Please re	eview and accept the following licenses
NVI	IDIA End User License Agreements
Taccept the terms of these licenses	
	Continu

言語選択

Welcome				
Asturianu	Bahasa Indonesia	Bosanski	Catal	là
Čeština	Cymraeg	Dansk	Deuts	ch
Eesti	English	Español	Espera	nto
Euskara	Français	Gaeilge	Galeg	o
Hrvatski	Íslenska	Italiano	Kurd	î
Latviski	Lietuviškai	Magyar	Nederla	inds
No localization (UTF-8)	Norsk bokmål	Norsk nynorsk	Polsk	ii
Português	Português do Brasil	Română	Sámegi	illii
			Back	Continue

English を選択し、[continue]ボタンを押し、次に進みます。

キーボードレイアウト選択

English(US)を左右のリストで選択します。日本語キーボードを選択すること も可能です。

	System Configuration
Keyboard layout	
Choose your keyboard layout:	
English (Nigeria)	English (US)
English (South Africa)	English (US) - Cherokee
English (UK)	English (US) - English (Colemak)
English (US)	English (US) - English (Dvorak)
Esperanto	English (US) - English (Dvorak, alt. intl.)
Estonian	English (US) - English (Dvorak, intl., with dead keys)
Faroese	English (US) - English (Dvorak, left-handed)
Type here to test your keyboard	
Detect Keyboard Layout	
	Back Continue
	the second s

タイムゾーン選択

地図上で Tokyo を選択し、[continue]ボタンを押し、次に進みます。



ユーザー名、パスワード、コンピュータ名設定

Your name が入力されると Pick a username にも自動的に同じ内容が入力さ れます。ログインするときは Pick a username が使用されます。 password は6文字以上で入力します。

1997 - C.		
Your name:	myjetson	4
Your computer's name:	myjetson-desktop 🚽	
	The name it uses when it talks to other computers.	
Pick a username:	myjetson 🧳	
Choose a password:	Fair password	
Confirm your password:	••••••	
	O Log in automatically	
	Require my password to log in	
		Back Continu

APP パーティションサイズの設定

APP パーティションサイズをMB単位で設定します。通常、初期状態の 最大サイズを設定します。0または空白入力は最大サイズとなります。 [continue]ボタンを押し、次に進みます。

		System Configuration			
APP Partition	Size				
	Please enter de: Default value in inpu Enter 0 or le	sired size of APP partition in ut field is the maximum size t eave blank to use the maximu	Megabytes (MB). that can be accepted. um size value.		
	Current size: 1	2273 MB, Maximum accepted	d size: 28497 MB		
28497					
				Back	Continue

設定は完了し、システム構成が開始されます。しばらくして、ログイン画面となります。 ログオンユーザー名を選択します。

パスワードを入力します。



- 5. コンピュータを使用するための基礎知識、設定、およびツールの導入
 - 1) 画面解像度の設定

13.3 インチ 1920x1080 液晶モニターを接続しているため、フルHDでの解像度に なっており、表示が小さいので、モニター解像度を下げ、1280x720(16:9)にします。 希望の解像度に設定してください。

起動画面左、縦にアプリケーションのアイコンが並んでいます。 このエリアで希望 アイコンをクリックすることで、即アプリケーションが実行できます。 このエリア を Ubuntu Dock と呼びます。

このエリアの System Setting > Displays 画面を開き、希望の解像度に設定します。

- 5) 画面スクリーンショットの保存 操作手順書作成等で、画面コピーが必要になる場合があります。以下の操作で画面 コピーが保存できます。

画面からの操作

Ubuntu Dock からアプリケーションの選択、Search your computer を選択します。 検索ワードに screenshot と入力します。入力するとカメラのアイコンで、 Screenshot アプリケーションが表示されるので、クリックします。スクリーンショ ット実行画面が表示されます。また同時に Ubuntu Dock にアイコンが表示されま す。以降、Ubuntu Dock に Screenshot アイコンを固定したい場合、アイコン上マ ウス右ボタンの Lock from Launcher をクリックします。

画面コピー対象を、画面全体、現在の指定ウインドウ、選択エリアから選択します。 「Take Screenshot」ボタンで実行します。次に保存画面となります。保存ファイル名、保存ホルダーを選択し、完了します。

キーボードからの操作

[Prt Sc]キー	>ディスクトップ全体
Alt + [Prt Sc] キー	>現在のウインドウ、有効に機能していない
Shift + [Prt Sc]キー	>選択ウインドウ

4)時刻の設定

ネットワークに接続していない場合、Ubuntu Dock から System Setting > Time & Date 画面を開き、マニュアルで時刻を合わせます。ネットワークに接続している 場合は、自動的に設定されます。

5) ターミナルの起動

Windows でのコマンドプロント同様のターミナルを起動し、左ディスクトップ Ubuntu Dock にセットさせます。

Ubuntu Dock から search your computer をクリックします。検索用キーワードの 入力画面となるので、「Terminal」と入力します。

関連のアプリケーションが表示されるので、このなかから「Terminal」をクリック します。

以下、ターミナルが起動されます。

この状態で Terminal アプリケーションが Ubuntu Dock 左ディスクトップに表示 されているので、このアイコン上、右クリック、「Lock to Launcher」をクリックし ます。以降、この Ubuntu Dock のアイコンでターミナルが起動できます。 6. ネットワークの設定

WIFI 接続セットアップ

まず、有線 LAN で動作を確認します。LAN コネクタに LAN ケーブルを接続し、ディ スクトップ上にあるブラウザ「Chromium web Browser」をクリックし、インターネッ ト接続が OK かどうか確認します。

次に先の購入リストからの WIFI 無線 LAN モジュール WN725N を USB に挿入しま す。認識され、画面に「ネットワークが有効になった」とのメッセージが表示されます。

次に Ubuntu Dock 左アプリケーションから「System settings」>[Network]を選択しま す。「Wireless」タブに有効な無線 LAN ルーターの SSID 一覧が表示されるので、希望 ルーターSSID をクリックします。チェックマークがオンされ、パスワードが要求され ます。ルーター製品に同封されている情報カードのパスワードを入力後、「Connect」ボ タンをクリックします。

接続 OK となり画面上、ツールバーに WIFI のアイコンが表示されます。有線 LAN ケ ーブルは不要ですので、外します。

SSH での接続

力します。

手持ちのなれている Windows パソコンから、Jetson Nano を操作できるよう SSH 接続 の設定をします。

まず、パソコン側の IP を確認します。Windows コマンドラインから「ipconfig」と入



次に、Jetson Nano の IP アドレスを確認します。画面上ツールバーの WIFI アイコンを クリック、表示されるプルダウンメニューで「Connection Information」をクリックし ます。現在の IP アドレスが表示されます。

ここで、Windows パソコンと Jetson Nano でのサブネットマスクが一致していること を確認します。異なっている場合は Winwos パソコン側で変更、一致させます。

Windows パソコン側でターミナルソフト Tera Term を起動します。インストールされ ていない場合は、事前にインストールしておきます。

Tera Term: 新しい接	続
© TCP/ĮP	ホスト(丁): 192.168.0.104 「ビヒストリ(Q) サービス: ○ Telnet TCPボート#(P): 22 ● <u>S</u> SH SSHバージョン(V): SSH2 ・ ● その他 プロトコル(Q): UNSPEC ・
● シリアル(<u>E</u>)	ポート(<u>R)</u> :
	OK キャンセル ヘルプ(出)

ここで、ホストに Jetson Nano の IP アドレスを入力します。TCP ボードは 22、サー ビスは SSH2 を選択します。セキュリティ警告ができますが、「続行」をクリックしま す。以下、先に登録した Jetson Nano ユーザー名、パスワードを入力します。

SSH認証		x
ログイン中: 192.168.0	0.104	
認証が必要です。		
ユーザ名(<u>N</u>):	myjetson	
バスフレーズ(<u>P</u>):	•••••	
	▼パスワードをメモリ上に記憶する(M)	
	□ エージェント転送する(0)	
◎ プレインテキス	トを使う(_)	
○ <u>R</u> SA/DSA/EC	DSA/ED25519鍵を使う 秘密鏈(<u>K)</u> :	
⊙ r <u>h</u> osts(SSH1)を	を使う ローカルのユーザ名(山):	
	ホスト鍵(E):	
◎ チャレンジレス	ボンス認証を使う(キーボードインタラクティブ)(C)	
Pageantを使う		
	OK 接続地向D)	

OK ボタンをクリックで、以下、ターミナルモードでログオンできます。



FTP での接続

SSH では、Windows パソコンと Jetson Nano 間でのファイルの受け渡しができません。 FTP サーバーで接続することで作業が効率化できます。

まず、オープンソースの FTP クライアントアプリケーション、WinSCP をパソコンに ダウンロード、インストールします。完了後、WinSCP を起動すると、以下のログイン

画面となります。 🖥 WinSCP ログイン 🚅 新しいサイト 📮 dlinano@19つ セッション dinano@192.168.0.104 dinano@192.168.1.7 root@203.138.223.155 転送プロトコル(E) SETP -ホスト名(日) ポート番号(R) 22 🚔 パスワード回 ユーザ名(U) 保存(5) 🔻 設定[0... ▼ 団 ログイン マ 閉じる ヘルプロ ツール(T) 🔻 管理(M) -21

ログイン画面で、ホスト名に、Jetson Nano の IP アドレス、ユーザー名に Jetson Nano のログインユーザー名、パスワードを入力します。

WinSCP ログイン 第50(1914)ト ↓ dinano@192,168.0,104 ↓ dinano@192,168.1,7 ↓ root@203,138.223,155	セッション 転送プロトコル(E) SFTP ホスト名(L) ユーザ名(L) パスワード(P) myjetson 「保存(S) ▼ 設定(D) ▼
ツールエ) ▼ 管理(M)	 ・ ・

「ログイン」ボタンクリックで、以下のファイル一覧画面となり、双方間でのファイル 転送が可能となります。

ᡖ Documents - myjetson@	⊉192.168.0. :	104 - WinSCP	1	A 1	_	-		
□-カル(<u>L</u>) マーク(<u>M</u>) コ	ファイル(E) ニ	コマンド(<u>C)</u> セッション	ン(<u>S</u>) オプション(<u>0</u>) リモート		プ(<u></u>)		
🖶 🚼 📚 同期(S) 🧊 g	2 💽 🐵	ער≠ 🔐 -(Q) א		転送設定 デフォルト		• 💋 •		
📮 myjetson@192.168.0.:	104 💣 新し	いセッション						
	(- 🗈 🗈 🏠 🖉	2	📙 myjets 👻 🚰 🔽	← - →	- 🗈 🖸 🏠 🖉	品 ファイルの検索(F)	2
🗐 アップロード(L) 👔 🗍	🦹 編集(E) 💃	× * +	»	■ ダウンロード(L) ■	2 編集	E) 🗙 🛃 🖓 プロパラ	≂न(P) 📴 🕞 💽	
C:¥Users¥yamada¥Docum	ents			/home/myjetson				
名前 拡張子	サイズ	種類	*	名前 拡張字		更新日時	パーミッション	所有者
🔒		ひとつ上のディ		🛃		2018/01/29 1:53:22	rwxr-xr-x	root
길 3dprinter		ファイル フォル		🌗 .cache		2020/01/29 15:52:21	rwx	myjet.
🔒 Apowersoft		ファイル フォル		🐌 .compiz		2018/01/29 2:11:05	rwx	myjet
Architecture		ファイル フォル		퉬 .config		2018/01/29 1:52:14	rwx	myjet.
\mu Arduino		ファイル フォル		퉬 .gnupg		2018/01/29 2:07:01	rwx	myjet.
🔒 Autodesk		ファイル フォル		퉬 .local		2018/01/29 2:07:04	rwx	myjet.
🔒 Aws		ファイル フォル		퉬 .nv		2018/01/29 2:07:12	rwx	myjet.
\mu BeagleBoneBlack		ファイル フォル] .pki		2018/01/29 1:10:15	rwx	myjet.
퉬 blender		ファイル フォル		퉬 Desktop		2018/01/29 2:07:10	rwxr-xr-x	myjet.
\mu Bluetooth		ファイル フォル) Documents		2018/01/29 2:07:08	rwxr-xr-x	myjet.
CatFootGoods		ファイル フォル	-) Downloads		2018/01/29 2:07:08	rwxr-xr-x	myjet.
 ₩ 		F.		•		m		•
0 B of 14,102 MiB in 0 of 28	32			0 B of 18,045 B in 0 of 2	1			
マークコマンド			_			6	SFTP-3 🖳	0:03:01

7. カメラのセットアップ

購入リストの MX219-D77 モジュール(Raspberry Pi カメラボード V2)を用意します。

Jetson Nano は MIPI-CSI カメラを標準でサポートしています。MIPI は Mobile Industry Proccessor Interface の略です。CSI は Camera Serial Interface の略です。このプロトコ ルは高速にカメラとの通信が可能です。USB スタックと異なってオーバーヘッドが少 ないです。CSI-2 カメラで最も普及しているのは、ラズベリーパイ用カメラ、 Raspberry Pi Camera Module V2 です。この Rpi カメラはソニー製イメージデバイス、 IMX-219 を使用しています。また赤外線カメラ、 Pi NoIR Camea も入手可能で、赤外 LED を照明として、ペアで使用することで暗闇での使用も可能です。注意事項として V1 Raspberry Pi Camera は使用できません。Jetson Nano のドライバーは VI のイメー ジデバイスを含んでいません。

インストールは簡単です、Jetson Nano が電源オフであることを確認し、コネクタ J13 に挿入します。コネクタの両サイドを持ちロック用キーを上に上げます。フラットケー ブルの極性に注意し、挿入します。最後にロックキーをカチと音がするまで、下ろしま す。

この状態では、カメラは、ボードの内部方向に向きます。外部方向に向ける場合、フラ ットケーブルをひねる必要があり、この場合、別途フラットケーブルの長いものを購入 する必要があります。アマゾン等で入手できると思います。またカメラ側も同様にケー ブルの抜き差しができます。

あと注意事項として、カメラには保護シールが貼ってあり、そのまま使用すると画像状 態が悪くなります。かならず外すようにして下さい。

カメラが正しく、認識されているかどうか確認します。ターミナルを起動し、以下入力 します。

ls /dev/video0 /dev/video と表示されればOKです。

Github に Jetson Nano 用カメラサンプルプログラム、インターフェイスが用意されて いるので、インストールします。以下、入力します。

git clone https://github.com/JetsonHacksNano/CSI-Camera

インストール後、CSI-Cameraのフォルダに移動し、パイソンプログラム、 Simple camera.py を起動します。

python simple_camera.py

以下、カメラ画像が表示されます。

また、すでにインストールされている Gstreamer コマンドでもカメラテストができます。

gst-launch-1.0 nvarguscamerasrc ! 'video/x-raw(memory:NVMM), width=1920, height=1080, format=(string)NV12, framerate=(fraction)30/1' ! nvoverlaysink -e

カメラ画像が表示されます。

USB カメラとしては logitech c920 webcam があげられますが、高速高解像度処理では CSI カメラが適しているといわれています。通常速度では、使いまわし、メーカーサポ ートの点から USB カメラがベストです。 推論パッケージ(jetson-inference)のインストール、ビルド Jetson Nanoのボード上のみで、TensorRT による推論、また Pytorch による転移学習 を実行させることができます。推論では、C++または Python による画像分類、物体検 出のアプリケーションを作成することができます。

以下の視覚基本機能が TensorNet の派生として用意されています。 画像分類(image recognition)は imageNet 物体検出(object localization)は detectNet セマンテイックセグメンテーション(semantic segmentation)は segNet

これら AI 機能を実行させるコンポーネントとしてまず、JetPack が必要です。ただ、 Jetson Nano セットアップ時の SD カードに含まれており、インストールは不要です。 JetPack は AI を開発し、展開するために、またコンピュータ視覚アプリケーション開 発のための SDK です。以下のコンポーネントが用意されています。

L4T Kernel/BSP(Linux for Tegra)

NVIDIA から NVIDIA Tegra シリーズ用に配布されている GNU/Linux ベー スのシステムソフトウエア。

CUDA Toolkit (Compute Unified Device Architecture)

NVIDIA から提供されている並列演算プラッフォーム、および API。CUDA 対応 GPU(graphics processing unit)で GPGPU(General -Purpose computing on Graphics Porcessing Units)処理が可能とするもの。

cuDNN(CUDA Deep neural Network library)

NVIDIA から提供されているディープニューラルネットワーク用の GPU 対応高速基本ライブラリ。cuDNN は以下の広く使用されているディープラーニングフレームワークを高速化します。

Caffe Caffe2 Chainer Keras MATLAB MxNet TensorFlow Pytorch TensorRT

TensorRT は、NVIDIA から提供されているディープラーニング推論のハイパ フォーマンス SDK です。ディープラーニング推論の最適化機能、精度補正機 能、ランタイムを含んでいます。通常の CPU に比べ 40 倍の高速処理が可能 です。TensorRT は CUDA をベースとして、多くのメジャーなフレームワー クで作成されたニューラルネットワーク学習モデルに対して最適化、高精度 化します。

OpenCV(Open source computer vision)

主に、リアルタイムコンピュータビジョン処理を目的とした、クロスプラット フォーム、オープンソースライブラリです。以下のディープラーニングフレー ムワークに対応しています。

TensorFlow

PvTorch

Caffe

VisionWorks

コンピュータビジョンアプリケーション Khronos OpenVX Standrd を実装、 拡張した、コンピュータビジョン、イメージ処理用ソフトウエア開発パッケー ジです。 CUDA 対応 GPU,SOC に最適化しています。

Multimedia API's

推論パッケージ(jetson-inference)のダウンロード、ビルド、インストール 多くの学習済み DNN(Deep Learning Network)モデルが自動的にダウンロードされ、 実行可能となります。Caffe、TensorFlow UFF、ONNX 対応であり、ユーザー作成の 学習済みモデルも対応しています。パッケージは GitHub から入手します。

まず、ダウンロードの前に、以下で git cmake のインストールを確実にします、 \$ sudo apt-get update \$ sudo apt-get install git cmake

次に、以下、クーロンを作成します。 \$ git clone https://github.com/dusty-nv/jetson-inference \$ cd jetson-inference \$ git submodule update --init Python 開発パッケージ

推論パッケージの Python 機能は、Python C API を使用したネイティブ C++コード結 合 Python 拡張モジュールにより実装されています。このパッケージを構築中、システ ムの Python のバージョンを検索し、存在する Python の各バージョン (Python2.7,3.6.3.7等)に対してバインディングを構築します。またインストールされて いる numpy のバージョンに対してもバインディングを構築します。

初期状態、Ubuntu には Python2.7 の libpython-dev および python-numpy がプリイン ストールされています。Ubuntu により Python3.6 はプリインストールされているのに もかかわらず、Python3.6 開発パッケージ、libpython3-dev、および python3-numpy は インストールされていません。しかし、これら開発パッケージは Python C API を構築 するバインディング時に必要となります。したがってこの推論パッケージで Python3.6 用

バインディングを作成したい場合、次のステップに移る前に、以下、このパッケージを インストールしてください。

\$ sudo apt-get install libpython3-dev python3-numpy

Cmake による構成実行

次にビルド用ディレクトリを作成し、cmake を実行します。Cmake が実行されるとき、 スクリプト(CmakePrebuild.sh)が起動され、すべての関連パッケージがインストールさ れ、DNN モデルがダウンロードされます。

\$ cd jetson-inference # omit if working directory is already jetson-inference/
from above

\$ mkdir build

\$ cd build

\$ cmake ../

このコマンドは、CmakePreBuild.sh スクリプトを起動します。スクリプトはいくつか の必須パッケージをイントールする sudo 特権を問い合わせてきます。また。このスク リプトはウエブサービスから学習済みネットワークをダウンロードします。

モデルのダインロード

推論パッケージは多くの学習済みネットワークを用意しています。モデルダウンロー ダー(dwonload-models.sh)で希望モデルを選択し、ダウンロード、インストールできま す。初期状態では、ディスクスペース節約のため、すべてのモデルがダウンロード選択 にされていません。ただ、希望のモデルが選択可能です。また、後日、任意でモデルダ ウンローダーを実行できます。 モデルをダウンロードするためには Box.com に接続する必要があり、この接続ができ ないユーザーは、ミラーサイトが以下に用意されてます。

https://github.com/dusty-nv/jetson-inference/releases

後日、モデルダウンローダーを再度実行させたい場合は、以下のコマンドで実行できま す。

\$ cd jetson-inference/tools

./download-models.sh

PyTorch のインストール

後で述べますが、転移学習でネットワークを再度、学習させたい場合、JetPack が 4.2 以降であれば、オプションで PyTorch というツールのインストールで可能となります。 これは任意のオプションです。転移学習のステップが不要であれば、PyTorch のインス トールは不要です。このステップをスキップしてください。

必要であれば、以下の画面で、Pytorch for Python 2.7、Pytorch for Python 3.6 を個別、 または同時に選択してください。不要な場合は選択をしないで、終了します。

自動 Pytorch インストーラーは JetPack4.2 以降が必要です。他のバージョンは <u>http://eLinux.org/Jetson_Zoo</u> を参照し、ソースからビルドしてください。

このツールも、後日、再度実行させたい場合は、以下のコマンドで実行できます。 \$ cd jetson-inference/build \$./install-pytorch.sh 推論パッケージのコンパイル
Jetson-inference/build ディレクトリにあることを確認します。ライブラリ、Python 拡張バインディング、サンプルコードを構築す るため、以下コマンドを実行します。
\$ cd jetson-inference/build #もし etson-inference/build であれば不要
\$ make
\$ sudo make install
\$ sudo ldconfig

jetson-inference/build/aarch64 フォルダーに以下のディレクトリ構成でビルドされます。

|-build

¥aarch64

ビルドされたサンプルバイナリ
ネットワークモデル
テスト用画像
ヘッダーファイル
ビルドされたライブラリ

バイナリファイル(build/aarch64/bin/), ヘッダーファイル(build/aarch64/include/), ライブラリファイ(build/aarch64/lib/)、これらはまた、sudo make install のコマンドス テップ時に、/usr/local/にもインストールされます。

また、sudo make install のコマンド実行時に jetson.inference と jetson.utils モジュー ルの Python へのバインディングが /usr/lib/python*/dist-packages/に実行保存さ れます。コードが変更された場合、再度実行する必要があります。 9. Jetson Nano オンボード機械学習、推論の実行

インストールされた推論パッケージ内で、物体検出のデモを実行させてみます。物体検 出は、各物体の領域ボックスを検出することによりフレーム内の位置を得るものです。 物体検出ネットワークはあるフレーム内の多くの異なる物体を検出することができま す。

物体検出ネットワーク、detectNet は入力されたイメージから、各物体の領域ボックス 座標、物体のクラス、推論精度のリストを出力するものです。detectNet は Python ま たは C++から利用可能です。以下、多くの学習済み物体検出 モデルがダウンロード、 利用できます。

利用可能な学習済み物体検出モデル

detectnet-console コマンドの引数、--network で使用する学習済みモデルを指定することができます。

--Model-- --CLI argument-- --NetworkType enum-- --Object classes--SSD-Mobilenet-v1 ssd-mobilenet-v1 SSD_MOBILENET_V1 91 (COCO classes) SSD-Mobilenet-v2 ssd-mobilenet-v2 SSD_MOBILENET_V2 91 (COCO classes) SSD-Inception-v2ssd-inception-v2 SSD INCEPTION V2 91 (COCO classes) DetectNet-COCO-Dog coco-dog COCO DOG dogs DetectNet-COCO-Bottle coco-bottle COCO BOTTLEbottles DetectNet-COCO-Chair coco-chair COCO_CHAIR chairs DetectNet-COCO-Airplane coco-airplaneCOCO AIRPLANE airplanes ped-100 pednet PEDNET pedestrians multiped-500multipedPEDNET_MULTI pedestrians, luggage facenet-120 facenet FACENET faces

追加でネットワークモデルをダウンロードする場合は、以下のツールを実行します。

\$ cd jetson-inference/tools

\$./download-models.sh

異なる物体検出モデルで実行する場合、コマンドラインの引数 --network フラグで 上記希望モデルの CLI arguments を指定します。初期値は SSD-Mobilenet-v2 で、こ のモデルを使用する場合 --network フラグの指定は不要です。 例えば、モデルダウンローダーツールで SSD-Inception-v2 をダウンロードした場合、 # C++

\$./detectnet-console --network=ssd-inception-v2 input.jpg output.jpg
Python

\$./detectnet-console.py --network=ssd-inception-v2 input.jpg output.jpg となります。

detectNet の使用例として、以下、C++、または Python コマンドラインインターフェ イスが用意されています。

detectnet-console.cpp (C++)

detectnet-console.py (Python)

またライブカメラでの物体検出として、C++または Python が用意されています。 detectnet-camera.cpp (C++) detectnet-camera.py (Python)

コマンドラインからの物体検出

detectnet-console プログラムは静止画像から物体の位置を検出します。

以下重要なコマンドラインパラメータです。

入力画像ファイルのパス(jpg, png, tga, bmp)

出力画像ファイルのパス、オプション(jpg, png, tga, bmp)

検出モデルの指定、オプション(ディフォルトは SSD-Mobilenet-v2)

オーバーレイフラグ、カンマで区切られたボックス、ラベル等、オプション

ディフォルトは--overlay=box,labels,conf でボックス、ラベル、検出精度値が 表示されます。

アルファ値、オーバーレイのブレンディング値、オプション、ディフィルトは 120 検出の最小値、オプション、ディフォルトは 0.5

これらは、カスタムモデルにおいても有効です。また、これらの詳細はアプリケー ショ ンを起動するとき、--help フラグで入手できます。または各言語のコードを参照して下 さい。 初期値の SSD-Mobilenet-v2 モデルでイメージファイルから歩行者を検出するサンプ ルが以下です。

C++

\$./detectnet-console --network=ssd-mobilenet-v2 images/peds_0.jpg output.jpg
#--network フラグはオプション

Python

\$./detectnet-console.py --network=ssd-mobilenet-v2 images/peds_0.jpg
output.jpg

#--network フラグはオプション

注意事項として、最初にモデルを実行させる場合、TensorRT はネットワーク最適化の ため、多少の時間を要します。この最適化ネットワークファイルはディスクにキャッシ ュされるので、以後は処理が早くなります。

SSD ベースモデル、91-class MS COCO データセットは人間、乗り物、動物、身の回りの各種の物を学習済みです。

テスト用の各種イメージ、cat_*.jpg, dog_*.jpg, horse_*.jpg, peds_*.jpg、等が images/ に用意されています。

ライブカメラ物体検出デモの実行

detectnet-camera.cpp (C++)

detectnet-camera.py (Python)

先の detectnet-console サンプルと同様、これらカメラアプリケーションは detection network を使用します。detectnet-camera では以下の各種のコマンドラインオプション を使用できます。

--network フラグは物体検出モデルを指定できます。ディフォルトは SSD-Mobilenetv2).です。

--overlay フラグは、カンマで区切られたボックス、ラベル、認識精度で、ディフォルトは--overlay=box,labels,conf でボックス、ラベル、検出精度値が表示されます。

--alpha value はアルファ値、オーバーレイのブレンディング値でオプションです。ディ フィルトは 120 --threshold value は検出の最小値、オプショで、ディフォルトは 0.5 です。 --camera フラグは使用カメラの指定です。

MIPI CSI カメラの場合、センサーインデックス(0 or 1, ect.)を指定します。V4L2 USB カメラは /dev/video ノード (/dev/video0, /dev/video1, ect.)を指定します。

ディフォルトは MIPI CSI sensor 0 (--camera=0)です。

--width、--height フラグはカメラの解像度で、ディフォルトは 1280x720 です。 解像度はカメラがサポートしている範囲で指定する必要があります。 以下のコマンドで可能なフォーマットを確認できます。

\$ sudo apt-get install v4l-utils

\$ v4l2-ctl --list-formats-ext

これらのフラグを組み合わせて使用できます。カスタムモデルの使用に有効な追加の パラメータもあります。また、これらの詳細はアプリケーケーションを起動するとき、 --help フラグで入手できます。または各言語のコードを参照して下さい。

以下、典型的なプログラム起動例です。

C++

\$./detectnet-camera

SSD-Mobilenet-v2の使用、初期値 MIPI CSI camera (1280x720)

\$./detectnet-camera --network=ssd-inception-v2

SSD-Inception-v2の使用、初期値 MIPI CSI camera (1280x720)

\$./detectnet-camera --camera=/dev/video0

SSD-Mobilenet-v2 の使用, V4L2 カメラ /dev/video0 (1280x720)

\$./detectnet-camera --width=640 --height=480

SSD-Mobilenet-v2の使用、初期値 MIPI CSI camera (640x480)

Python
\$./detectnet-camera.py
SSD-Mobilenet-v2 の使用、初期値 MIPI CSI camera (1280x720)
\$./detectnet-camera.pynetwork=ssd-inception-v2
#SSD-Inception-v2 の使用、初期値 MIPI CSI camera (1280x720)
\$./detectnet-camera.pycamera=/dev/video0
#SSD-Mobilenet-v2 の使用, V4L2 カメラ /dev/video0 (1280x720)
\$./detectnet-camera.pywidth=640height=480
SSD-Mobilenet-v2 の使用,、初期値 MIPI CSI camera (640x480)
使用なカメラ例は以下 Jetson Wiki にあります。
https://eLinux.org/Jetson_Nano#Cameras
Allied Vision MIPI CSI-2 (one open-source CSI-2 driver for all cameras on
Github.com)
Alvium 1500 C-050 0.5MP PYTHON 480
Alvium 1500 C-120 1.2MP AR0135CS
Alvium 1500 C-210 2.1MP AR0521
Alvium 1500 C-500 5MP AR0521
Alvium 1800 C-040 0.4MP Sony IMX287
Alvium 1800 C-158 1.6MP Sony IMX273
Allied Vision USB3 Vision
Alvium 1800 U-040 0.4MP Sony IMX287
Alvium 1800 U-050 0.5MP PYTHON 480
Alvium 1800 U-120 1.2MP AR0135CS
Alvium 1800 U-158 1.6MP Sony IMX273
Alvium 1800 U-500 5MP AR0521
Alvium 1800 U-501m NIR 5MP AR0522
e-con Systems e-CAM50_CUNANO (5MP AR0521 MIPI Camera)
e-con Systems e-CAM30_CUNANO (3.4 MP MIPI Camera)
e-con Systems STEEReoCAM™ (2.0 MP MIPI Stereo Camera)
Logitech C270 (USB webcam)
Logitech C920 (USB webcam)
Leopard Imaging LI-IMX219-MIPI-FF-NANO (IMX219 sensor)
Raspberry Pi Camera v2 (IMX219 sensor)
Appro AP-IMX179-MIPIx1 (IMX179 sensor)
Appro AP-IMX290-MIPIx1 (IMX290 sensor)

Stereolabs ZED (stereo camera)

検出された物体の領域ボックスでオーバーレイされたライブカメラ動画が OpenGL ウ インドウに表示されます。現時点では SSD ベースモデルが最も高いパフォーマンスを 示します。以下、 COCO-DOG モデルを使用した場合です。 # C++

\$./detectnet-camera --network=coco-dog

Python

\$./detectnet-camera.py --network=coco-dog

もし、ビデオストリームに望む物体が検出されない場合、または間違った物体が検出さ れた場合は、検出閾値 –threshold を大きくするか、または小さくするか調整してくだ さい。初期値は 0.5 です。