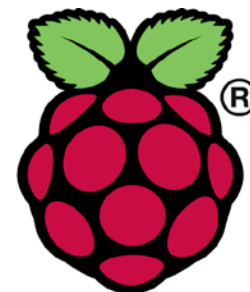


抜粋版

はじめてのAIプログラム学習キット5

USBアクセラセータを使ってTensorflow liteの基礎マスター、体験

TF lite学習編(Coral版)



Raspberry Pi

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

	ページ	
• RaspberryPi運用マニュアル		
• RaspberryPiについて	4	
• Linux基本コマンド	5	
• RaspberryPi基本操作	6	
• 日常運用(ウイルススキャン、更新)	7	
• 学習キット 全体像	9	抜粋版のためページと 本文は異なります
• ハード概要	10	
• ソフト概要	12	
• Coral		
1. 概要	14	
2. Pycoral基本動作	15	
• Tensorflow lite		
1. ガイド		
① 概要	18	
② モデル選択	20	
③ モデル変換	21	
④ デバイスへディプロイ	24	
⑤ モデル最適化	25	
⑥ パフォーマンス	27	
⑦ TF lite API	28	

学習キット 目次

Tensorflow lite

ページ

2. 教材

- ① 画像分類(転移学習) [29](#)
- ② 画像分類(Picamera_tpu未使用) [32](#)
- ③ 画像分類(Picamera_tpu使用) [33](#)
- ④ 物体検出(Picamera) [34](#)
- ⑤ テキスト分類 [35](#)

3. 事例

- 一覧 [37](#)
- ① 画像分類 [38](#)
- ② 物体検出 [39](#)
- ③ ポーズ推定 [40](#)
- ④ 身体セグメント [41](#)
- ⑤ 物体追尾 [42](#)
- ⑥ 物体セグメント [43](#)
- ⑦ キーワード認識ゲーム [44](#)
- ⑧ 画像合成 [45](#)
- ⑨ 高解像度画像作成 [46](#)

RaspberryPi運用マニュアル

1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。
LinuxベースのRasbianOSで動作しております。

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、menu>shutdown>reboot; 左上のメニューから

- 終了: # shutdown

又は、menu>shutdown>shutdown; 左上のメニューから

- ログアウト # exit

又は、menu>shutdown>logout; 左上のメニューから

- **日本語／英語の入力切替**: キーボードのCTLとjを同時に押します(コントロール: 左下とj)



Raspberry Pi

Pi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

root@:~\$ cd /root/Documents ディレクトリの切り替え
 root@:/root/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ーして操作します)
 root@:~# 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
 root@:~# 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
 root@:~# 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

root@:~\$ su - スーパーユーザ(root)に切り替え、パスワードを入力
 root@:~# ps a 現状の動いているプロセスを表示
 root@:~# kill 特定のプロセスを強制終了
 root@:~# apt-get install pkg パッケージのインストールなどに使用
 root@:~# date 日付、時間の設定を行います。
 root@:~# leafpad /etc/network/interfaces インタフェースに記述している内容を変更します。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

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

pi@raspberrypi:~\$ sudo
 も同様のコマンドになります

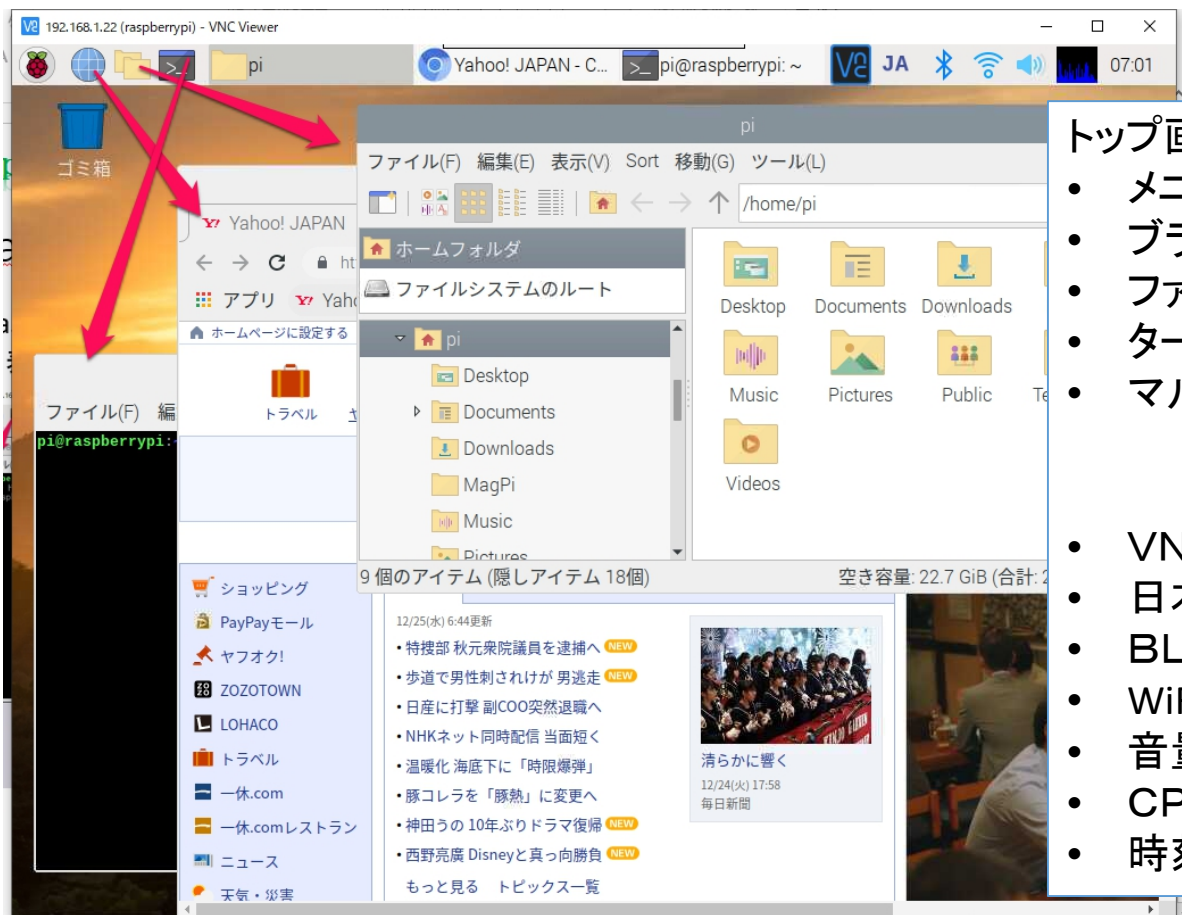


RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

① 表示画面と内容

デスクトップ上によく使うコマンド.txtがあります。
コピーして使ってください



トップ画面(上段のタスクバーで選択)

- メニュー
- ブラウザ
- ファイルマネージャ
- ターミナル
- マルチ画面選択

- VNC
- 日本語入力
- BLE
- WiFi
- 音量
- CPU使用率
- 時刻



RaspberryPi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います。

パターンファイル更新

手動スキャン時に更新されます

手動でスキャン

\$ sudo clamscan --infected --remove --recursive
自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後約30分位かかります。

```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
ERROR: /var/log/clamav/freshclam.log is locked by another  
ERROR: Problem with internal logger (UpdateLogFile = /var/  
og).  
root@raspberrypi: ~# leafpad /etc/clamav/freshclam.conf  
root@raspberrypi: ~# freshclam  
ClamAV update process started at Fri J  
main.cvd is up to date (version: 57, sigs: 4218790, f-level: 60, builder: mishh  
ammer)  
daily.cvd is up to date (version: 21862, sigs: 394456, f-level: 63, builder: neo  
)  
bytecode.cvd is up to date (version: 283, sigs: 53, f-level: 63, builder: neo)  
root@raspberrypi: ~# clamscan --infected --remove --recursive  
SCAN SUMMARY  
Known viruses: 4607906  
Engine version: 0.99.2  
Scanned directories: 264  
Scanned files: 2063  
Infected files: 0  
Data scanned: 61.31 MB  
Data read: 49.02 MB (ratio 1.25:1)  
Time: 71.844 sec (1 m 11 s)  
root@raspberrypi: ~#
```

手動でスキャン

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 to 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。

```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
Mem:          925      596      329           9        28       381  
-/+ buffers/cache:    186     739  
Swap:         99        0        99  
root@raspberrypi: ~# apt-get update  
ヒット http://archive.raspberrypi.org/jessie/contrib Translation-jp_JP  
ヒット http://archive.raspberrypi.org/jessie/main Translation-jp_JP  
ヒット http://mirrordirector.raspbian.org/jessie/contrib Translation-jp_JP  
ヒット http://mirrordirector.raspbian.org/jessie/main Translation-jp_JP  
ヒット http://mirrordirector.raspbian.org/jessie/contrib Translation-en  
ヒット http://mirrordirector.raspbian.org/jessie/main Translation-en  
無視 http://archive.raspberrypi.org/jessie/non-free Translation-jp_JP  
無視 http://archive.raspberrypi.org/jessie/non-free Translation-en  
無視 http://archive.raspberrypi.org/jessie/rpi Translation-en  
root@raspberrypi: ~# apt-get upgrade  
パッケージリストを読み込んでいます... 完了  
状態情報を読み取っています... 完了  
アップグレードパッケージを検出しています ... 以下のパッケージが自動でインストールされましたが、もう必要とされていません:  
libwebRTC-audio-processing-0 rtkit  
これを削除するには 'apt-get autoremove' を利用してください。  
アップグレード: 0 個、新規インストール: 0 個、削除: 0 個、保留: 0 個。  
root@raspberrypi: ~#
```

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

必ず実施前にバックアップ

学習キット 全体像(Coral版)

Coral版

ハードウェア

Raspberry Pi4B



+

AI用物品追加



カメラ
(オプション)



USB accelerator
: Google Coral

OS



Rasbian OS

AI系


TensorFlow
(v1,v2)

 TensorFlow.js

 PyTorch

画像系



YOLO

(オプション)

プログラム言語



モバイル


TensorFlow Lite

ハード概要

本体

品名	項目	内容	備考
Raspberry Pi 4 model B	CPU	1.5GHz 64bit クアッドコアCoretex-A72	
	GPU	デュアルコア VideoCore VI® 500MHz	
	メモリ	4GB RAM	
	OS	Raspbian buster(Debianベース)	
	インターフェース	2.4/5GHz WiFi(802.11 bgnac), Bluetooth 5.0, BLE, 1G ether, USB 2.0x2, USB 3.0x2, micro HDMIx2, microSDカード, 40 GPIO pin	Bluetooth 5.0, USB 3.0追加, microHDMIへ
	電源／消費電力	Micro USB Type C 3.0A	Type Cへ
	サイズ	85x56x18mm	
付属品		内容	備考
ケース		透明、白、黒から選択できます。	
microSD 128GB		Raspbian OS, 必要なモジュールをインストールして提供します。お客様が設定するものは必要最低限のパスワード設定、WiFi設定、VNC設定になります。	
プログラム		Tensorflow, tfjs, Tflite, Pytorch, 関連pip インストール済, Tensorflow, tfjs, Python サンプルプログラム	TfliteはUSBアクセラレータ購入時のみ提供
マニュアル		学習キット設定編、Python学習編、Tensorflow_v1,v2学習編、Tensorflow.js学習編、Pytorch学習編	

USB電源ケーブル、HDMIケーブルは付属していません。

別途オプション品を購入ください

ハード概要 オプション品

オプション品	内容	備考
Webカメラ	HD 1080p USB インターフェース モニタ上部に搭載可能	
バッテリー(3.0A 10000mAH)	Raspberry Piを携帯型で使用するためのバッテリーになります。本体は、3.0A以上の電流を使用します。	提供時期により提供する型番は変更になることがあります
電源ケーブル(Micro USB TypeC)1m	本体を動作させるためのmicro USBケーブルになります。基本部には付属しておりません。	
マイクロHDMIケーブル1.5m	Raspberry Piの初期設定時にモニタを接続するHDMIケーブル。2回目からはVNC接続をするため、使用しません。	
USBアクセラレータ	Google coralを提供します。事例も併せて提供します。Tensorflow liteで動作します。	



ソフト概要

①ソフトウェア一覧

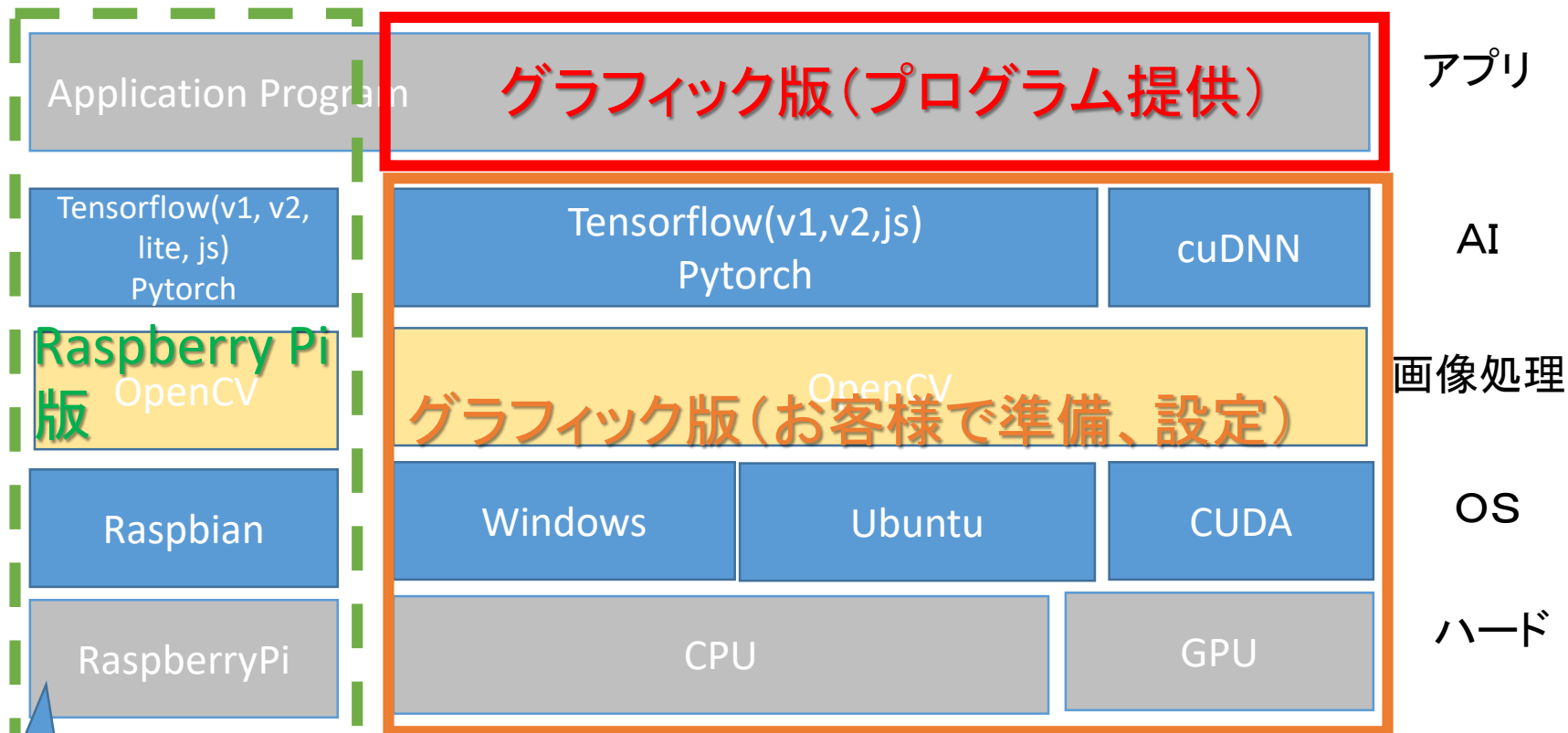
提供するソフトウェアの概要です。

区分	ソフト名	バージョン	備考
OS	Raspbian	buster	
プログラム言語	python3	3.7.3 3.6.9	仮想化で利用します。
	python2	2.7.15+	
	Node.js	V14.15	
AI用プログラム	tensorflow	2.4.0 1.14.0	Google
	Tensorflow.js	1.7.4	Google
	Tensorflow lite	2.5.0	Google coralと合わせて使用します
	pycoral	1.0.1	
	Pytorch	1.9	Facebook
サンプルプログラム	Mnist,cifar10など		使用AI言語により比較が可能
その他	Jupyter notebook、matplotlib など多数のpipライブラリ		

ソフト概要

②プロトコルスタック

プログラム言語:Python3, JavaScript



USBアクセラレータ(coral)

本格利用には、GPU版(グラフィック版)をお勧めします。学習の計算速度が、Raspberry Pi版に比べて、実測で5倍速から150倍速とストレスなく学習できます。

Coral

1. 概要

公式サイト

<https://coral.ai/>

① 概要

- Coralは、組み込みデバイスでニューラルネットワークを高速化するための完全なプラットフォームを提供します。アクセラレータの中心は、EdgeTPUコプロセッサです。これは、高性能のニューラルネット推論を提供する、小さいながらも強力な低電力ASICです。
- USBアクセラレータとして、今回は提供します。画像認識で30fpsまで高速処理できます。

② 利用方法

- Githubからインストールし、学習する: <https://github.com/google-coral/pycoral.git>
- 仮想化環境では、エラーが発生しました(不明)。Systemのpython3.7で動作します。

Coral

2. Pycoral基本動作

① 画像分類

<https://coral.ai/docs/accelerator/get-started/#3-run-a-model-on-the-edge-tpu>

- Pycoralの動作確認のために、画像分類試験です。

```
$ cd /home/pi/Documents/coral/pycoral
```

```
$ python3 examples/classify_image.py ¥
```

```
--model test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite ¥
```

```
--labels test_data/inat_bird_labels.txt ¥
```

```
--input test_data/parrot.jpg
```

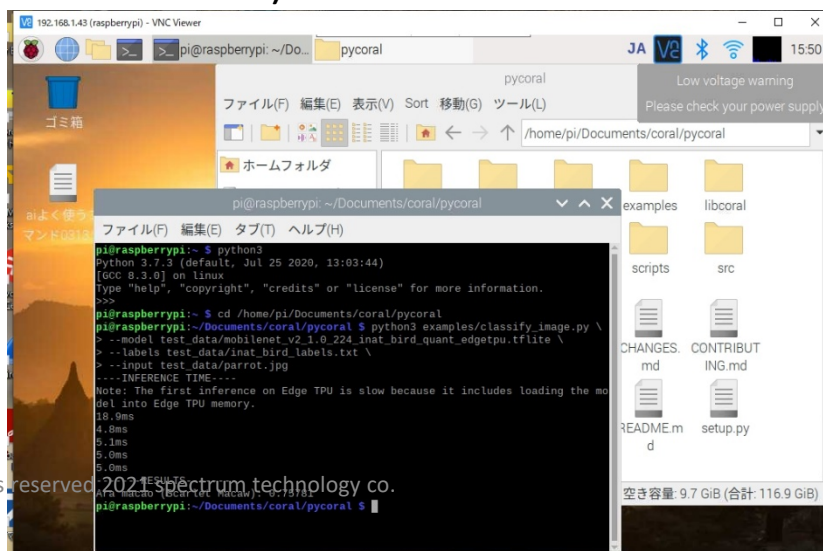
- 画像分類して、Ara macao (Scarlet Macaw): 0.75781と出力されたらOK。



入力コマンド

```
$ cd /home/pi/Documents/coral/pycoral
$ python3 examples/classify_image.py ¥
--model
test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite ¥
--labels test_data/inat_bird_labels.txt ¥
--input test_data/parrot.jpg
```

Coral



Coral

2. Pycoral基本動作

② Webカメラ(gstreamer)

<https://github.com/google-coral/examples-camera/tree/master/gstreamer>

- PycoralのWebカメラの物体分類、識別試験。

```
$ cd /home/pi/Documents/coral/examples-camera/gstreamer
```

```
$ python3 classify.py
```

分類結果が画面に出力

```
$ python3 detect.py
```

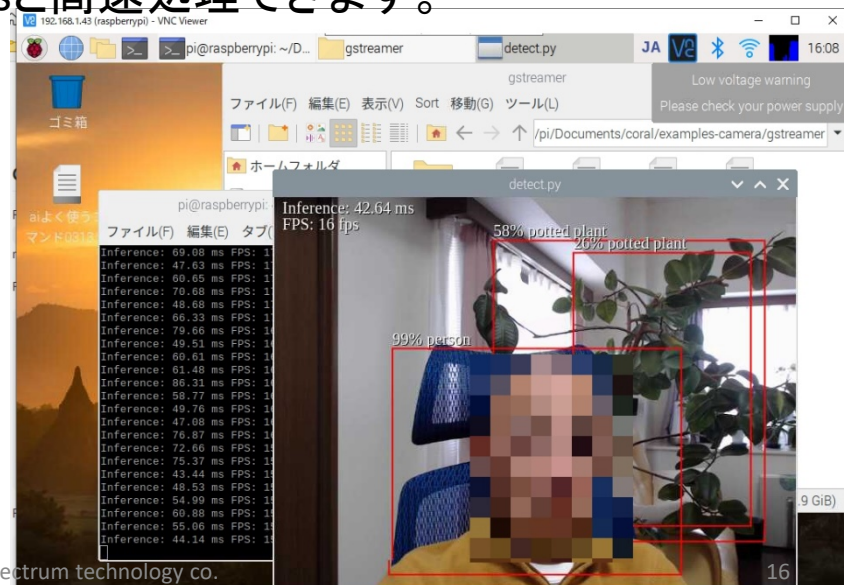
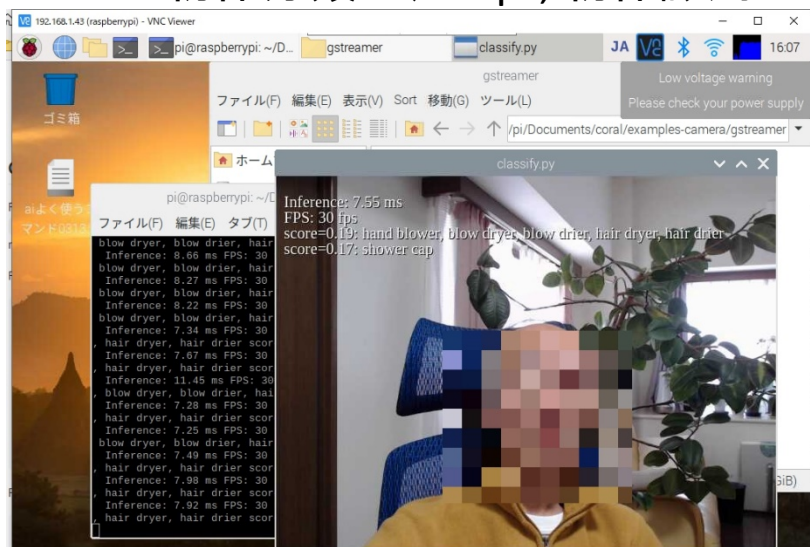
識別した結果をgstreamerを使って場所と物体、確率を表示

- 物体分類で、30fps, 物体識別で17fpsと高速処理できます。

入力コマンド

```
$ cd /home/pi/Documents/coral/examples-camera/gstreamer
$ python3 classify.py
$ python3 detect.py
```

Coral



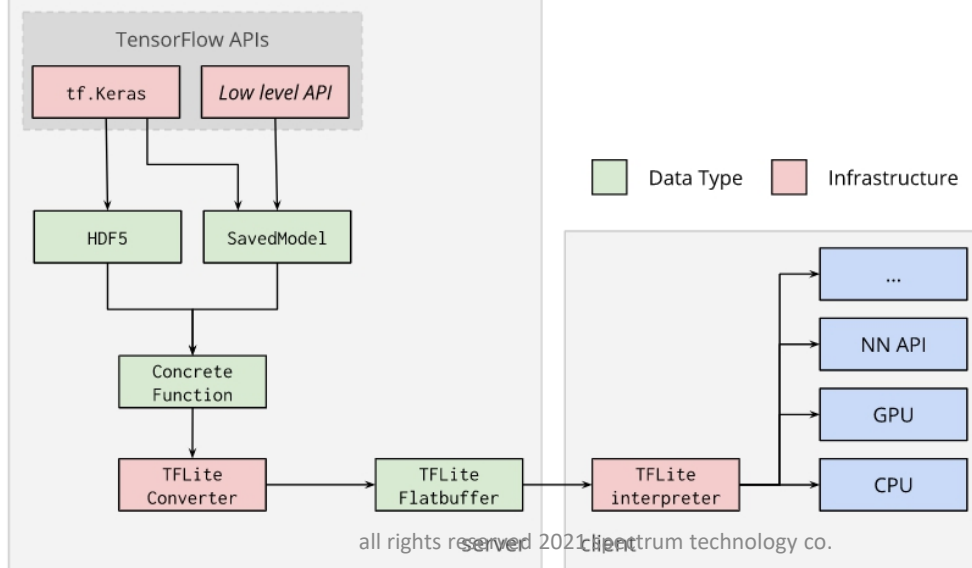
Tensorflow lite

1. ガイド

① 概要

- TensorFlow Lite は、Googleが開発したtensorflowのうち、モバイル、IoT用デバイス向けに作成されました。低遅延、軽量のモデルで動作します。
- TensorFlow Lite は、デバイス上で TensorFlow モデルを実行するためのオープンソースのディープラーニングフレームワークです。TensorFlow Lite を初めて使用する場合は、まず事前トレーニング済みモデルを試し、実際のデバイスで以下のサンプルアプリを実行して、TensorFlow Lite で何ができるかを確認することをおすすめします。
- 全体の概念図は下図のとおり。

server側は、
colab等になります。



client側は、
coral+raspberrypi
になります

Tensorflow lite

1. ガイド

① 概要

- <https://www.tensorflow.org/lite/guide?hl=ja>
- Tfliteの使い方の説明を上記で記載
- エッジ側での機械学習の特徴
 - レイテンシ: サーバーとの往復の通信にかかる時間が発生しない
 - プライバシー: データをデバイスから外部に送信する必要がない
 - 接続: インターネット接続が必要ない
 - 消費電力: ネットワーク接続に電力を大量に消費する必要がない
- エッジデバイスには、Android,RaspberryPi、マイコンなど各種あります。
- 開発ワークフロー



TfconverterでTfliteに変換

モデルサイズを最小化

TFモデルを学習済等から選択

TfinterpreterとTfliteのファイルを
エッジデバイスにディプロイ

Tensorflow lite

1. ガイド

② モデル選択

- https://www.tensorflow.org/lite/guide/hosted_models?hl=ja
- 事前学習モデルなどを選択できます。
 - 画像分類
 - 物体検出
 - 画像セグメンテーション等
- 事例で別途詳細に、解説してゆきます。

画像分類

画像分類の詳細については、[画像分類](#)を参照してください。わずかな行数のコードで[画像分類モデルを統合](#)するには、TensorFlow Lite Task Libraryをご覧ください。

量子化モデル

[量子化画像分類モデル](#)は、精度を低くして、最小のモデルサイズと最速のパフォーマンスを提供します。パフォーマンス値は、Android 10 が搭載された Pixel 3 で測定されています。

TensorFlow Hub には多くの[量子化モデル](#)が提供されているため、TensorFlow Hub からさらに詳しいモデル情報を取得できます。

モデル名	論文とモデル	モデルサイズ	トップ1の精度	トップ5の精度	CPU、4 スレッド	NNAPI
Mobilenet_V1_0.25_128_quant	論文 、 tfLite&pb	0.5 Mb	39.5%	64.4%	0.8 ms	2 ms
Mobilenet_V1_0.25_160_quant	論文 、 tfLite&pb	0.5 Mb	42.8%	68.1%	1.3 ms	2.4 ms
Mobilenet_V1_0.25_192_quant	論文 、 tfLite&pb	0.5 Mb	45.7%	70.8%	1.8 ms	2.6 ms
Mobilenet_V1_0.25_224_quant	論文 、 tfLite&pb	0.5 Mb	48.2%	72.8%	2.3 ms	2.9 ms
Mobilenet_V1_0.50_128_quant	論文 、 tfLite&pb	1.4 Mb	54.9%	78.1%	1.7 ms	2.6 ms
Mobilenet_V1_0.50_160_quant	論文 、 tfLite&pb	1.4 Mb	57.2%	80.5%	2.6 ms	2.9 ms
Mobilenet_V1_0.50_192_quant	論文 、 tfLite&pb	1.4 Mb	59.9%	82.1%	3.6 ms	3.3 ms
Mobilenet_V1_0.50_224_quant	論文 、 tfLite&pb	1.4 Mb	61.2%	83.2%	4.7 ms	3.6 ms
Mobilenet_V1_0.75_128_quant	論文 、 tfLite&pb	2.6 Mb	55.9%	79.1%	3.1 ms	3.2 ms

Tensorflow lite

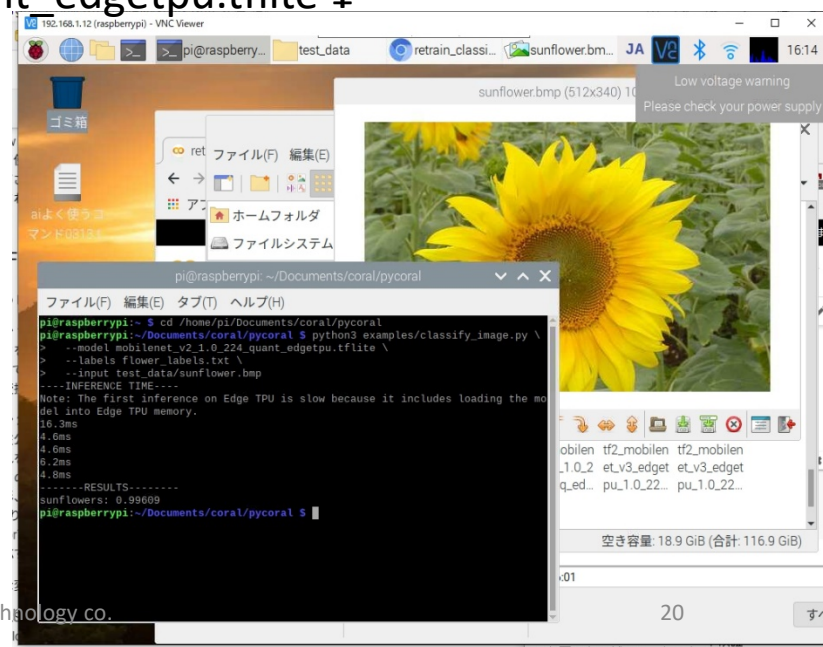
1. ガイド

④ デバイスにディプロイ 花の例

- https://colab.research.google.com/github/google-coral/tutorials/blob/master/retrain_classification_ptq_tf1.ipynb#scrollTo=hRTa3Ee15WsJ
- RaspberryPiでTPU(Coral)を搭載して試験します。
 - `cd /home/pi/Documents/coral/pycoral`
 - `python3 examples/classify_image.py ¥`
 - `--model mobilenet_v2_1.0_224_quant_edgetpu.tflite ¥`
 - `--labels flower_labels.txt ¥`
 - `--input test_data/sunflower.bmp`
- 結果: 99%の確率でsunflower。
- 他のテストデータでも実施してください。

```
cd /home/pi/Documents/coral/pycoral
python3 examples/classify_image.py ¥
--model mobilenet_v2_1.0_224_quant_edgetpu.tflite ¥
--labels flower_labels.txt ¥
--input test_data/sunflower.bmp
```

ダウンロードした2つのファイルは、既に右のフォルダに移動しています。



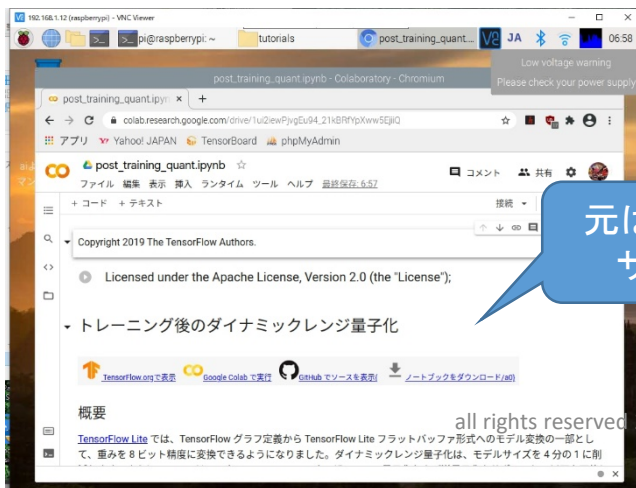
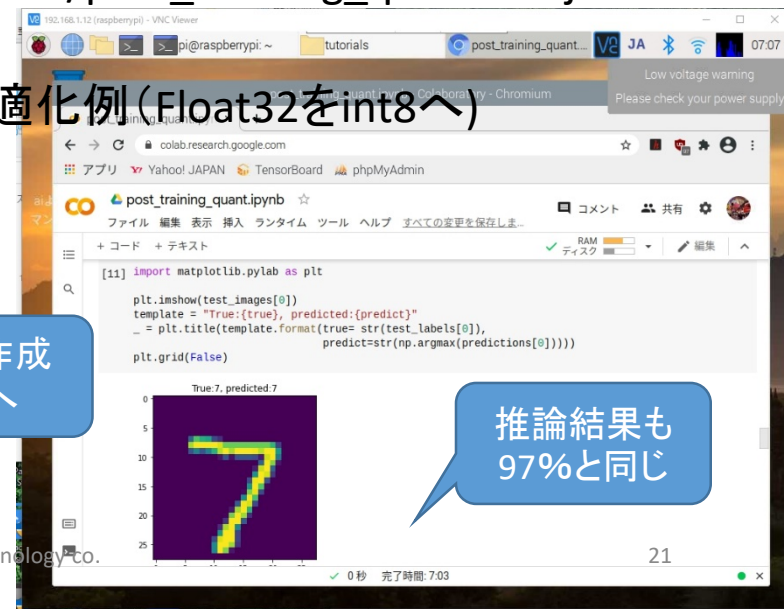
Tensorflow lite

1. ガイド

⑤ モデルの最適化 mnistダイナミックレンジ量子化例

- https://www.tensorflow.org/lite/performance/model_optimization?hl=ja
- 目的: Edge デバイスではメモリや計算能力が限られていることがよくあります。モデルにさまざまな最適化を適用すると、これらの制約内で実行できるようになります。
 - サイズ縮小、伝送遅延縮小、アクセラレータによる高速化、トレードオフ
- 最適化方法
 - 量子化、プルーニング、クラスタリング
- https://www.tensorflow.org/lite/performance/post_training_quant?hl=ja
- /home/pi/Documents/coral/tutorials
- post_training_quant.ipynb

mnistの最適化例(Float32をint8へ)

元はFloat32で作成
サイズが1/4へ推論結果も
97%と同じ

Tensorflow lite

1. ガイド

⑦ TF lite API一覧

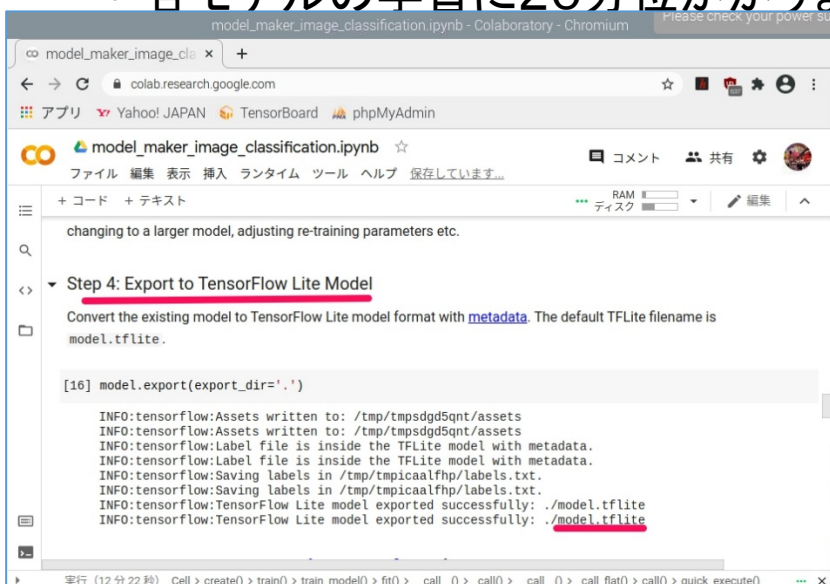
- https://www.tensorflow.org/lite/api_docs/python/tf/lite
- モデル変換からエッジデバイスで使う関数までの一覧を確認してください。
- Tflite converter: Colabを使ってモデル変換に使用
- Tflite interpreter: エッジデバイス(RaspberryPi)でconverterで変換したTfliteのモデルを使用
- Optimizer:Tfconverterで当初変換したモデルを量子化を使ってサイズを縮小し、エッジデバイス用に最適化します。

Tensorflow lite

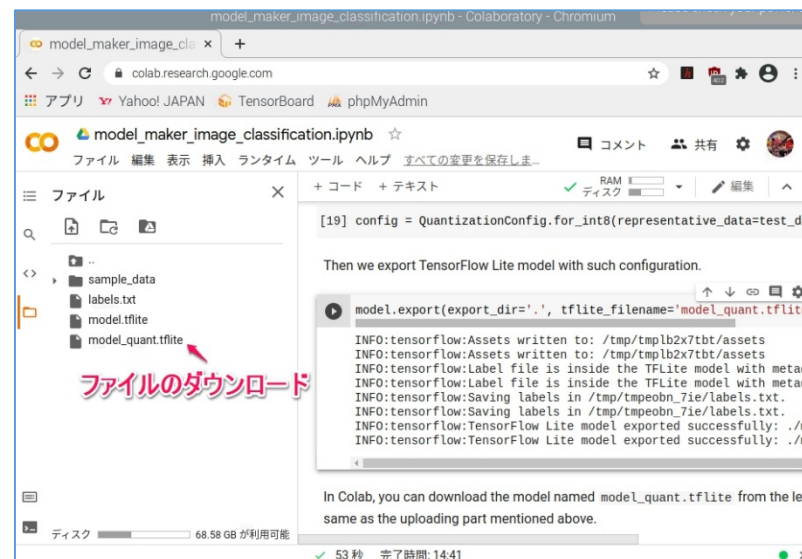
2. 教材

① 画像分類(転移学習)

- https://www.tensorflow.org/lite/tutorials/model_maker_image_classification?hl=ja
- 花の画像を分類するモデルで、Tfliteにエクスポートする全体の流れをcolabで行います。Google colabはクラウド上でGPUを搭載し、無料で利用できます。
- TF lite modelの出力が出ると完了
- 左のタグからローカルにダウンロードして、Pi+Coralのローカルで実施。別途説明
- 各モデルの学習に20分位かかります。



The screenshot shows a Google Colab notebook titled 'model_maker_image_classification.ipynb'. The code cell is at 'Step 4: Export to TensorFlow Lite Model'. It contains the command `model.export(export_dir='.')` and its output, which shows the model being successfully exported to `model.tflite`. The file explorer on the left shows the file `model.tflite` in the current directory.



The screenshot shows a Google Colab notebook titled 'model_maker_image_classification.ipynb'. The code cell is at 'Step 4: Export to TensorFlow Lite Model'. It contains the command `model.export(export_dir='.')` and its output, which shows the model being successfully exported to `model.tflite`. The file explorer on the left shows the file `model.tflite` in the current directory. A red arrow points to the file `model.tflite` in the file explorer, with the text 'ファイルのダウンロード' (File download) written next to it.

Tensorflow lite

2. 教材

② 画像分類(Picamera_tpu未使用)

- https://github.com/tensorflow/examples/tree/master/lite/examples/image_classification/raspberry_pi
- RaspberryPi搭載のPi cameraで画像分類を行います。
- cd
/home/pi/Documents/coral/examples/lite/examples/image_classification/raspberry_pi
- bash download.sh /tmp 関連の学習済モデルなどをtmpにダウンロード
- python3 classify_picamera.py ¥
--model /tmp/mobilenet_v1_1.0_224_quant.tflite ¥
--labels /tmp/labels_mobilenet_quant_v1_224.txt

```
$ cd
/home/pi/Documents/coral/examples/lite/examples/i
mage_classification/raspberry_pi
$ bash download.sh /tmp
$ python3 classify_picamera.py ¥
--model /tmp/mobilenet_v1_1.0_224_quant.tflite ¥
--labels /tmp/labels_mobilenet_quant_v1_224.txt
```

vnc接続ではカメラ
動画表示されませ
ん。Hdmiで



Tensorflow lite

3. 事例

一覧

A:難しい、B:やや難しい、C:普通

項番	項目	内容	難易度	習得時間(分)	備考
①	画像分類	写真から登録されているカテゴリ(鳥)で分類	C	10	
②	物体検出	写真から登録されている物体を検出します	C	10	
③	ポーズ推定	主要な体の関節の位置を推定することにより、画像や動画内の人物のポーズを推定	C	10	
④	身体セグメント	人物の身体を部位毎にセグメンテーションします	B	30	
⑤	物体追尾	個別の人物など追尾します。ID付与し個別に識別	B	30	
⑥	物体セグメント	物体をセグメンテーション(輪郭縁取り)します	B	30	
⑦	キーワード認識ゲーム	キーワードでへびを動かしてイチゴ?を食べるゲーム	B	60	
⑧	画像合成	二つの画像を合成する。Tensorflow.jsのインシュタインの例と同じ	C	20	
⑨	高解像度画像作成	低解像度の対応物から高解像度(HR)画像を復元する	B	30	

Tensorflow lite

3. 事例

① 画像分類

- 写真から登録されているカテゴリ(鳥)で分類します。
- こちらの[ページ](#)で紹介しました。
- `cd /home/pi/Documents/coral/pycoral`
- `python3 examples/classify_image.py ¥`
`--model test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite ¥`
`--labels test_data/inat_bird_labels.txt ¥`
`--input test_data/parrot.jpg`

入カコマンド

```
$ cd /home/pi/Documents/coral/pycoral
$ python3 examples/classify_image.py ¥
--model
test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite ¥
--labels test_data/inat_bird_labels.txt ¥
--input test_data/parrot.jpg
```



Figure 1. parrot.jpg

Tensorflow lite

3. 事例

⑦ キーワード認識ゲーム

- <https://github.com/google-coral/project-keyword-spotter>
- キーワードでヘビを動かしてイチゴ？を食べるゲーム。
- `cd /home/pi/Documents/coral/project-keyword-spotter`
- `sh install_requirements.sh`
- `python3 run_model.py` 指示内容をしゃべって音声認識させる
 - イチゴを食べるゲームの指示内容を認識させる
 - Turn up, turn down, turn left, turn rightとしゃべって認識させる訓練を実施
- `python3 run_hearing_snake.py` ヘビを言葉で指示してイチゴを食べるゲーム

入カコマンド

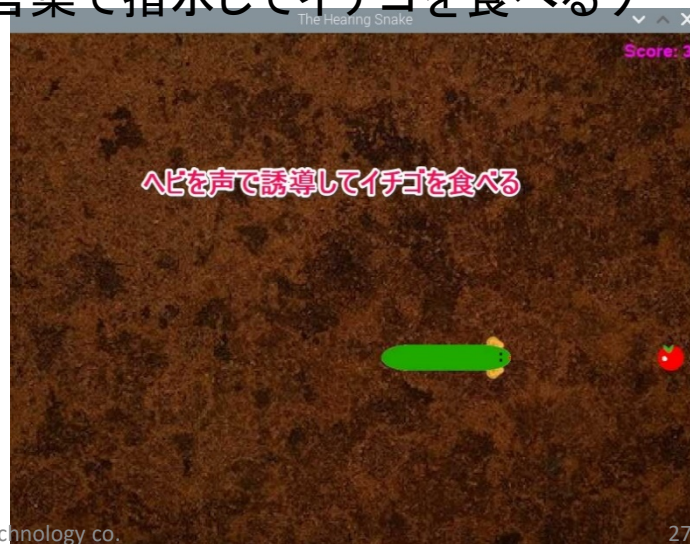
```
$ cd /home/pi/Documents/coral/project-keyword-spotter
$ sh install_requirements.sh
$ python3 run_model.py
$ python3 run_hearing_snake.py
```

マイク又はカメラを
接続してください

```
pi@raspberrypi: ~/Documents/coral/project-keyword-spotter
ファイル(F) 編集(E) タブ(T) ヘルプ(H)

negative (0.945)      no (0.039)
negative (0.941)      no (0.039)
negative (0.977)      no (0.008)
negative (0.992)
negative (0.988)
negative (0.980)      no (0.008)
negative (0.961)      turn_down (0.020)
negative (0.973)
negative (0.996)
negative (0.996)
negative (0.996)
negative (0.996)
negative (0.977)
negative (0.938)      turn_up (0.039)      turn_down (0.008)
turn_up* (0.965)      negative (0.020)      volume_up (0.008)
turn_up* (0.668)      negative (0.301)      volume_up (0.012)
negative (0.038)      move_up (0.020)      turn_up (0.016)
negative (0.973)      stop_game (0.020)
negative (0.953)      go_left (0.008)      channel_ten (0.008)
negative (0.957)      go_right (0.008)      go_down (0.008)
negative (0.973)
negative (0.980)      go_right (0.008)
negative (0.984)      go_right (0.008)
negative (0.996)
```

認識した場合



Tensorflow lite

3. 事例

⑧ 画像合成

- https://www.tensorflow.org/lite/examples/style_transfer/overview?hl=ja
- 二つの画像を合成する。Tensorflow.jsのアインシュタインの例と同じ
- /home/pi/Documents/coral/examples/lite/examples/style_transfer
- overview.ipynb
- Colabで実施

