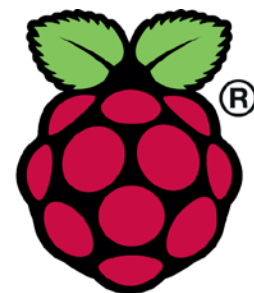


抜粋版

はじめてのMediapipe用AI開発キット

～ 最新のRaspberry Pi5を使い、google開発の物体、人物認識用AIソフトを有効活用～
開発編 (Pi5版)



Raspberry Pi

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales1@spectrum-tech.co.jp

開発編 目次

• Pi運用マニュアル	ページ
1. Piについて	4
2. 基本コマンド	4
3. 基本操作	6
4. 日常運用	7
• 開発キット 全体像	9
• ハード・ソフトウェア概要	
1. ハード概要	10
2. ソフト概要	
① ソフト一覧	11
② プログラム一覧	12
• Mediapipe	13
1. 画像映像	
① 物体検出	14
② 画像分類	20
③ 画像セグメンテーション	25
④ 相互セグメンテーション	31
⑤ ジェスチャー認識	36
⑥ 手ランドマーク検出	41
⑦ 画像埋め込み	47
⑧ 顔検出	52
⑨ 顔ランドマーク検出	57
⑩ ポーズランドマーク検出	63
⑪ 顔スタイル化	69
⑫ 総合ランドマーク検出	74

抜粋版のためページは一致しません

開発編 目次

2. テキスト

- ① テキスト分類 [80](#)
- ② テキスト埋め込み [84](#)
- ③ 言語検出 [88](#)

3. 音声

- ① 音声分類 [92](#)

参考

スマホ開発(android, ios)



Pi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、アプリケーション>ログアウト>再起動; 左上のメニューから
- 終了: # shutdown
又は、アプリケーション>ログアウト>シャットダウン; 左上のメニューから
- ログアウト # logout
又は、アプリケーション>ログアウト>ログアウト; 左上のメニューから
- 日本語／英語の入力切替: キーボードの全角/半角キーで切替ます。



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@:~# mousepad /etc/network/interfaces インタフェースに記述内容を変更します。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

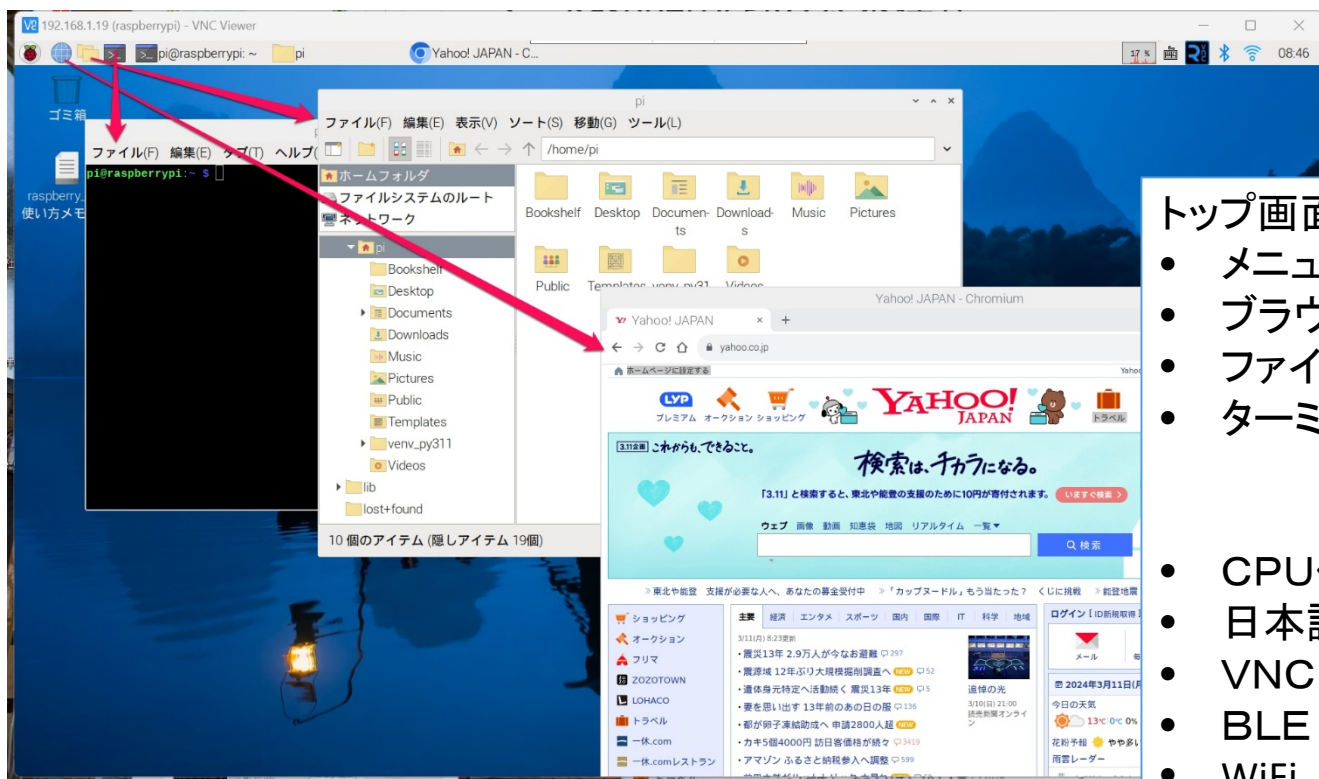


Raspberry Pi

RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

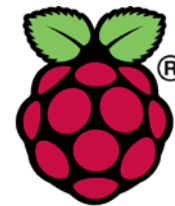
① 表示画面と内容



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

- CPU使用率
- 日本語入力
- VNC
- BLE
- WiFi
- 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10分位かかります。

```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~$ sudo clamscan --infected --remove --recursive  
----- SCAN SUMMARY -----  
Known viruses: 8686504  
Engine version: 1.0.3  
Scanned directories: 4406  
Scanned files: 20705  
Infected files: 0  
Data scanned: 2863.69 MB  
Data read: 2616.18 MB (ratio 1.09:1)  
Time: 570.533 sec (9 m 30 s)  
Start Date: 2024:03:11 08:50:19  
End Date: 2024:03:11 08:59:49  
pi@raspberrypi:~$
```



RaspberryPi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の実施してください。
- 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)  
End Date: 2024:03:11 08:59:49  
pi@raspberrypi:~$ sudo apt-get update  
ヒット:1 http://deb.debian.org/debian bookworm InRelease  
ヒット:2 http://deb.debian.org/debian-security bookworm-security InRelease  
ヒット:3 http://deb.debian.org/debian bookworm-updates InRelease  
ヒット:4 http://archive.raspberrypi.com/debian bookworm InRelease  
パッケージリストを読み込んでいます... 完了  
pi@raspberrypi:~$ sudo apt-get upgrade  
パッケージリストを読み込んでいます... 完了  
依存関係ツリーを作成しています... 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... 完了  
以下のパッケージが自動でインストールされましたが、もう必要とされません:  
ca-certificates-java default-jdk-headless default-jre default-jre-headless  
fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni  
libssl1.1 openjdk-17-jdk openjdk-17-jdk-headless openjdk-17-jre  
openjdk-17-jre-headless  
これを削除するには 'sudo apt autoremove' を利用してください。  
以下のパッケージは保留されます:  
libcamera-ipa libcamera-tools libpipewire-0.3-0 libpipewire-0.3-modules  
libspa-0.2-bluetooth libspa-0.2-modules pipewire pipewire-bin  
pipewire-libcamera pipewire-pulse python3-libcamera rpikam-apps  
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: 12 個。  
pi@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
```


開発キット 全体像(Pi5版)

ハードウェア

Raspberry Pi5



+

物品追加



電源アダプタ
(オプション)



HDMIケーブル
(オプション)



カメラ
(オプション)

OS



Rasbian OS

物体・人物認識

MediaPipe

MediaPipe Studio

CODEPEN

画像系



プログラム言語

python



AI系

PyTorch

弊社提供

1. ハード概要

本体

品名	項目	内容	備考
Raspberry Pi5	CPU	2.4GHz 4コア Cortex-A76 (ARMv8、64bit)	
	GPU	VideoCore VII®	
	メモリ	4GB RAM	
	OS	Raspbian bookworm(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	
	電源／消費電力	Micro USB Type C 3.0A	
	サイズ	85x56x18mm	
付属品		内容	備考
ケース		赤白、FAN付。	
microSD 64GB		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設定になります。	
プログラム		Mediapipe, 関連pip インストール済, Pythonサンプルプログラム多数	
マニュアル		開発キット 設定編、開発編	

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

2. ソフト概要

①ソフト一覧

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	Bookworm 64bit	
画像	Opencv	4.9.0	仮想化時
	codepen		Web版
物体・人物認識	mediapipe	0.10.9	
	Mediapipe studio		Web版
プログラム言語	python3	3.11.2	仮想化で利用
	javascript		Web版
AI用プログラム	Pytorch	2.1.2	
その他	Jupyter notebook、 matplotlibなど多数のpipライブラリ		

2. ソフトウェア概要

②. 開発キットプログラム一覧

A:実用可能
B:要確認
C:試験段階
X:エラー

id	区分	タスク名	概要	Mediapip e studio	Jupyter: 写真	python:写 真	python:カ メラ	python:ピ デオ(弊社 独自)	web:写真、 カメラ (codepen)	評価	コメント
1	画像映像系	物体検出	画像、映像から物体を検出	●	●	●	●		●	A	
2		画像分類	画像から物体等の種類ごとに分類します。	●	●	●	○:pi用		●	A	
3		画像セグメンテーション	画像から、物体を切り抜きセグメンテーションします	●	●	●	●	●	●	A	
4		相互セグメンテーション	画像をセグメンテーションし、一部の選択した画像の修正、ぼかし、差替えなどができます	●	●	●			▲写真のみ	A	
5		ジェスチャー認識	画像、映像からそのジェスチャーを認識します	●	●	●	○:pi用		●	A	fps=20超える、高速
6		手ランドマーク検出	画像、映像から手の詳細な位置を検出します	●	●	●	●	●	●	B	fps=10、普通
7		画像埋め込み	画像を数値化して、編集・加工に使用します	●	●	●			●	A	
8		顔検出	画像、映像から顔の位置を検出します	●	●	●	○:pi用		●	A	
9		顔ランドマーク検出	画像、映像から顔の詳細位置(目、口、眉など)を検出します	●	●	●	●	●	●	B	fps=10、普通
10		ポーズランドマーク検出	画像、映像からポーズの詳細位置(頭、肩、手など)を検出します	●	●	●	●	●	●	C	fps=4、遅い
11		顔スタイル化	画像、映像から顔のアバターを作成します。	●	×エラー	×エラー			●	C	一部エラー
12		総合ランドマーク検出	手、顔、ポーズのランドマークを検出します				●	●		C	かなり遅い
13	テキスト系	テキスト分類	入力したテキストからポジティブ、ネガティブの分類を行います	●テキスト	●テキスト	●テキスト			●テキスト	A	
14		テキスト埋め込み	テキストを数値化して、編集・加工に使用します	●テキスト	●テキスト	●テキスト			●テキスト	A	
15		言語検出	テキストから使用している言語を検出します	●テキスト	●テキスト	●テキスト			●テキスト	A	
16	音声系	音声分類	音声から話し声、TV、音楽、環境音などをリアルタイムで分類します	●音声	●音声	●音声			●音声	A	マイクからの入力
17	生成AI系	画像生成	テキストから画像生成を行います。								未検証
18		LLM推論	テキストから生成し、推論してゆきます								未検証

Mediapipe

- MediaPipe Solutions は、人工知能 (AI) および機械学習 (ML) 技術を実アプリケーションに迅速に適用するためのライブラリとツールのスイートを提供します。これらのソリューションを実アプリケーションにすぐに組み込んで、ニーズに合わせてカスタマイズし、複数の開発プラットフォームで使用できます。MediaPipe Solutions は MediaPipe オープンソースプロジェクトの一部であるため、アプリケーションのニーズに合わせてソリューションコードをさらにカスタマイズできます。
- Mediapipe <https://developers.google.com/mediapipe>
- Github: <https://github.com/google/mediapipe>
- 特徴
 - 利用が簡単
 - セルフサービスの ML ソリューション。ローコード API またはノーコードスタジオを使用してカスタマイズ、評価、プロトタイプ、デプロイを行う
 - 革新的
 - Google ML の専門知識を活用して作成された、一般的なタスク向けの高度な ML ソリューション。
 - 高速処理
 - ハードウェア アクセラレーションを含むエンドツーエンドの最適化を実現しながら、バッテリー駆動のデバイスでも十分に動作するほど高速です。

1. 画像映像

①. 物体検出

画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

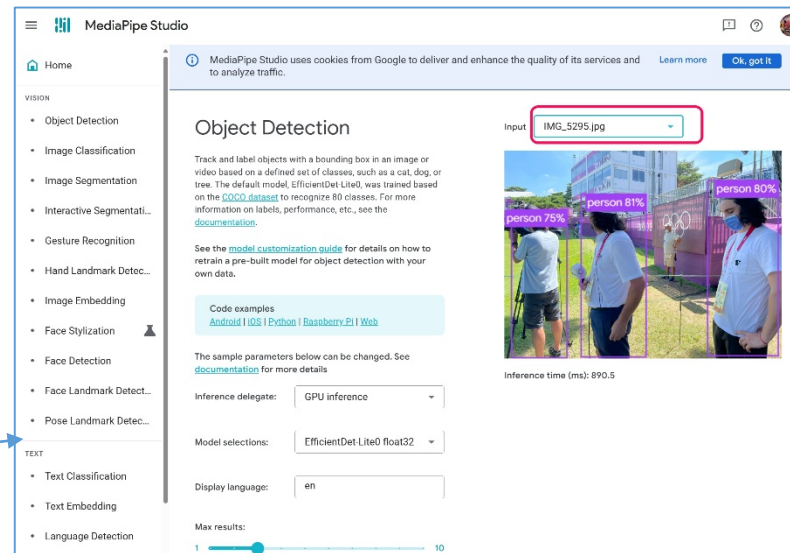
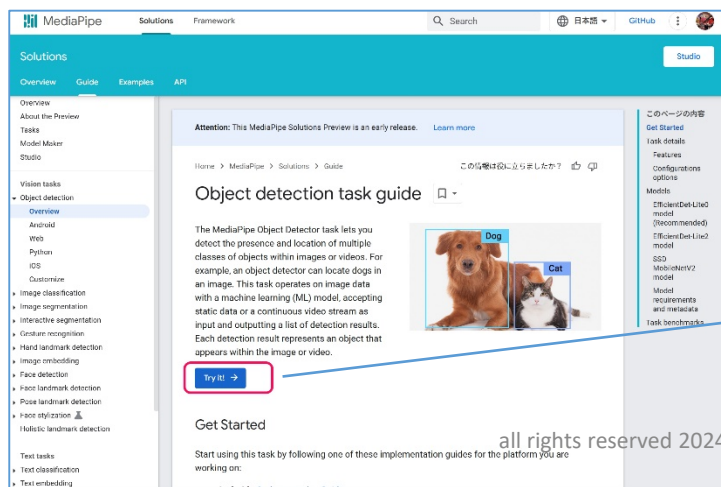
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

A) Mediapipe studio

https://mediapipe-studio.webapps.google.com/demo/object_detector

使用方法

- Mediapipeのサイトからtry itを押すと別画面に起動
- inputに写真をアップロード
- 自動的に認識し、区分と確率がでます。



1. 画像映像

①. 物体検出

画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

B) Jupyter notebook

使用方法

仮想化で使用

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/notebook
```

```
$ jupyter notebook
```

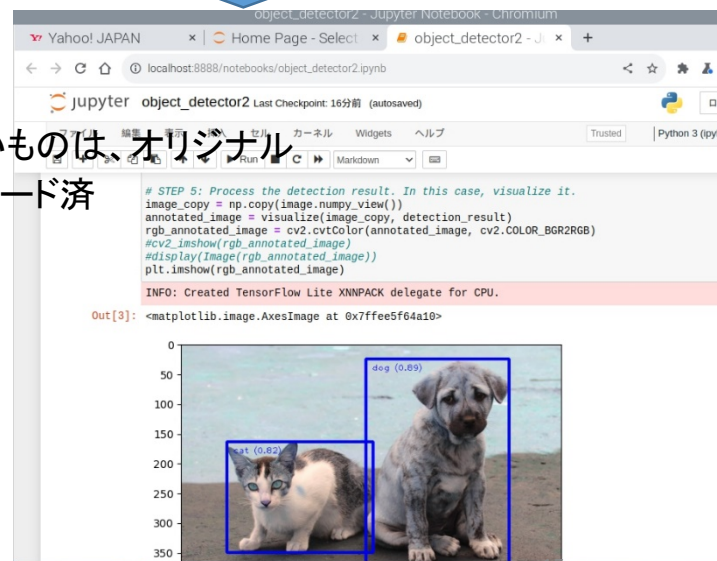
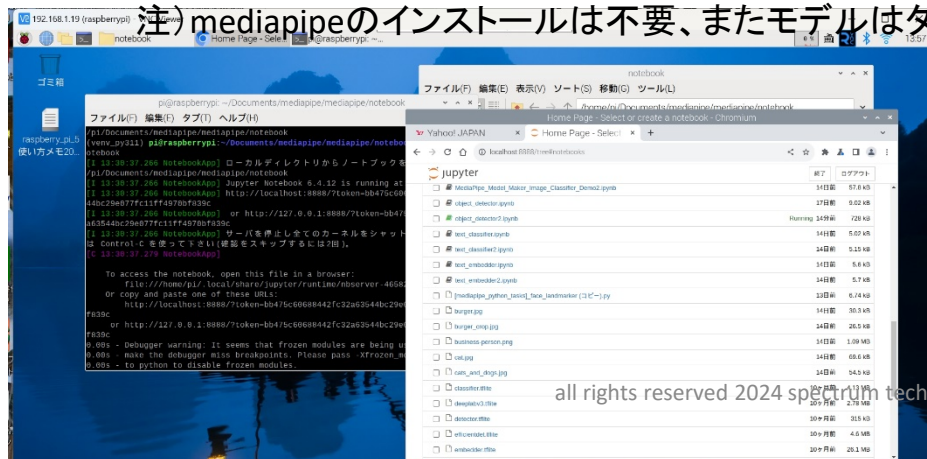
object_detector2.ipynb を選択してクリック。2がついていないものは、オリジナル

注) mediapipeのインストールは不要、またモデルはダウンロード済

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
$ cd
/home/pi/Documents/mediapipe/mediapipe/
notebook
$ jupyter notebook
object_detector2.ipynb
```

vnc接続では、jupyter notebookは遅い。HDMIで実施推奨



1. 画像映像

①. 物体検出

画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

C) python:画像

使用方法

仮想化で使用

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/examples/python/object_detection
```

```
$ python object_detector.py
```

- Jupyterと同様に結果の写真のみポップアップ

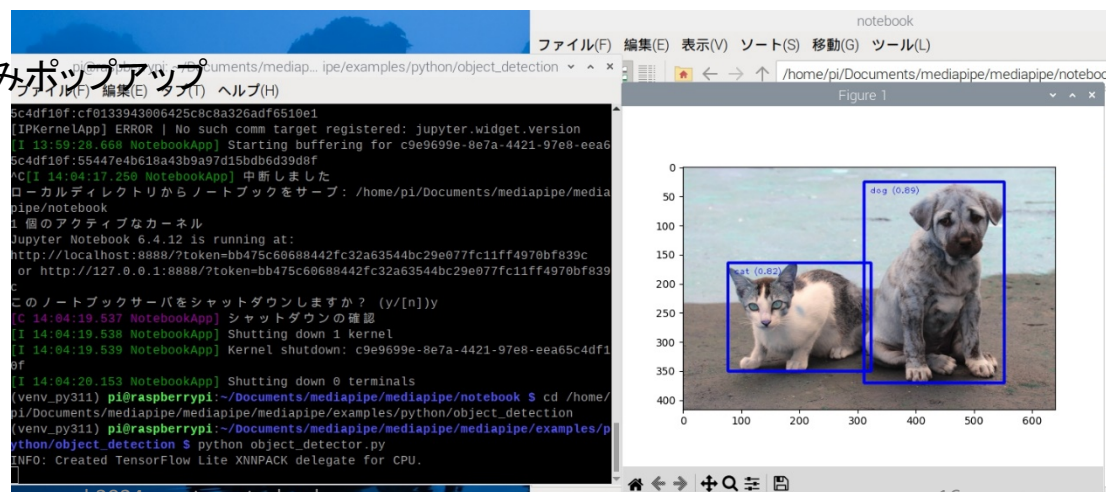
コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd
```

```
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/object_detection
```

```
$ python object_detector.py
```



画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
 - B) Jupyter notebook:pythonのプログラムで、順番に内容確認します。
 - C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
 - D) Web:codepen:html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ
- C) python:カメラ (usbカメラでテスト、piカメラは未検証)

使用方法

仮想化で使用

```
$ source /home/pi/venv py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/object_detection
```

```
$ python detect.py
```

- カメラで物体検出します。
- かなり高速処理が可能

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
$ cd
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/object_detection
$ python detect.py
```



1. 画像映像

①. 物体検出

画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

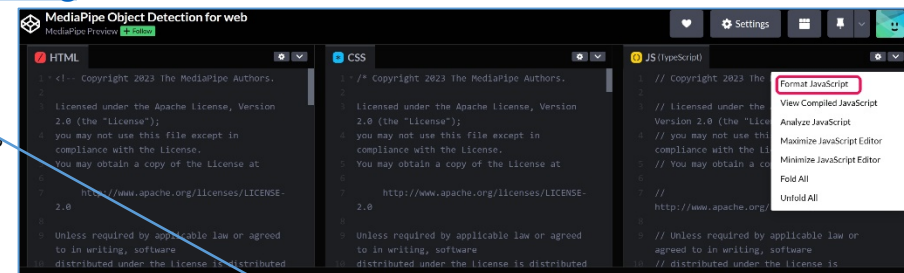
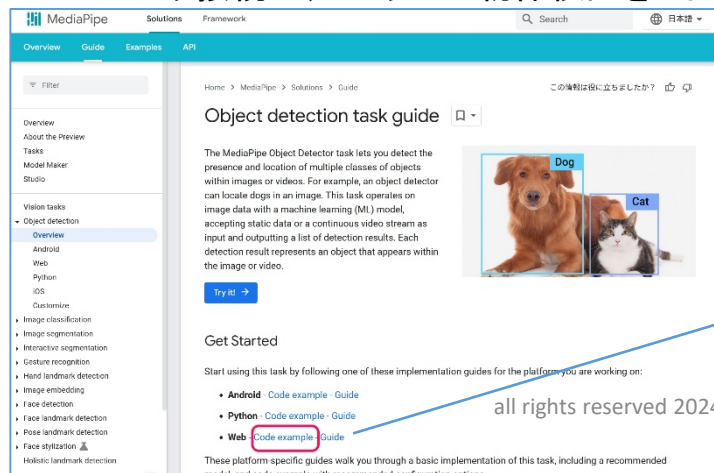
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

D) codepen

使用方法

<https://codepen.io/mediapipe-preview/pen/vYrWvNg>

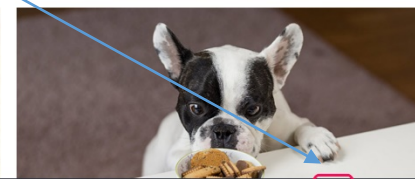
- Forkで自分のサイトへ取り込み
- format javascriptを押すと、起動
- 写真のサンプルを押すと、物体を検出します。
- カメラ接続で、カメラでの物体検出をします。



Multiple object detection using the MediaPipe Object Detector task

Demo: Detecting Images

Click on an image below to detect objects in the image.



1. 画像映像

①. 物体検出

画像、動画から物体を検出します。

https://developers.google.com/mediapipe/solutions/vision/object_detector

方法

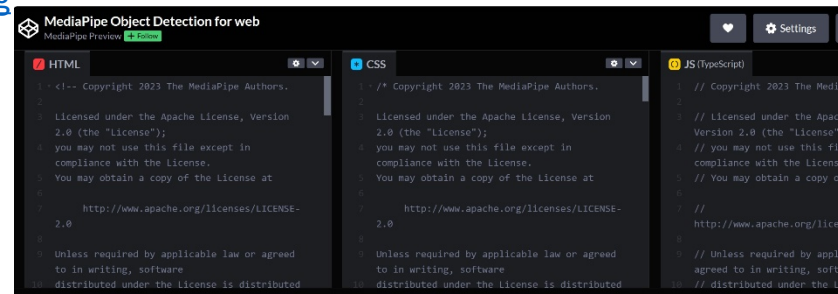
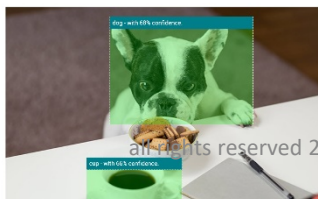
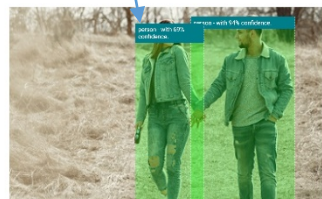
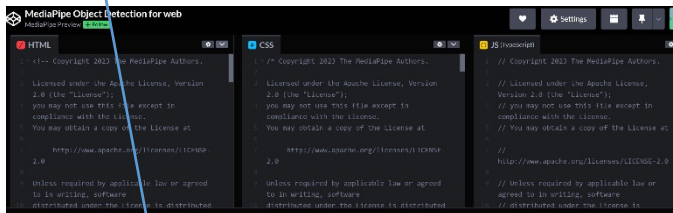
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

D) codepen

使用方法

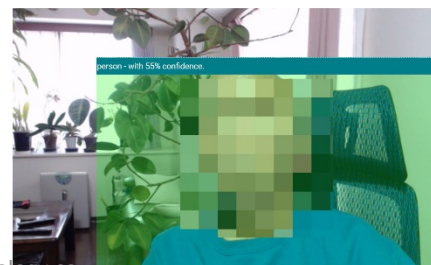
<https://codepen.io/mediapipe-preview/pen/vYrWvNg>

- Forkで自分のサイトへ取り込み
- format javascriptを押すと、起動
- 写真のサンプルを押すと、物体を検出します。
- カメラ接続で、カメラでの物体検出をします。



Hold some objects up close to your webcam to get a real-time detection! When ready click 'enable webcam' below and accept access to the webcam.

This demo uses a model trained on the COCO dataset. It can identify 80 different classes of object in an image. [See a list of available classes](#)



enable cameraを押す

1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

A) Mediapipe studio

https://mediapipe-studio.webapps.google.com/demo/face_landmarker

使用方法

- Mediapipeのサイトからtry itを押すと別画面に起動
- カメラを選択、顔のランドマークを検出します。

The screenshot displays the MediaPipe Studio web application. The top navigation bar includes 'MediaPipe', 'Solutions', and 'Framework'. The left sidebar lists various vision tasks, with 'Face landmark detection' selected. The main content area is titled 'Face landmark detection guide' and features a video player showing a man's face with blue landmark points. Below the video, there are two images of the same man, one with landmarks and one without. The right sidebar contains a list of tasks, with 'Face Landmark Detect...' selected. The bottom section shows the 'Face Landmark Detection' demo interface, which includes an input field for choosing an image file, a video player showing a man's face with landmarks, and a list of code examples (Android, iOS, Python, Raspberry Pi, Web). The demo also displays inference time (671.8 ms) and a note about switching between Grid and Avatar renderers.

all rights reserved 2024 spectrum technology co.

1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

B) Jupyter notebook

使用方法

仮想化で使用

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/notebook
```

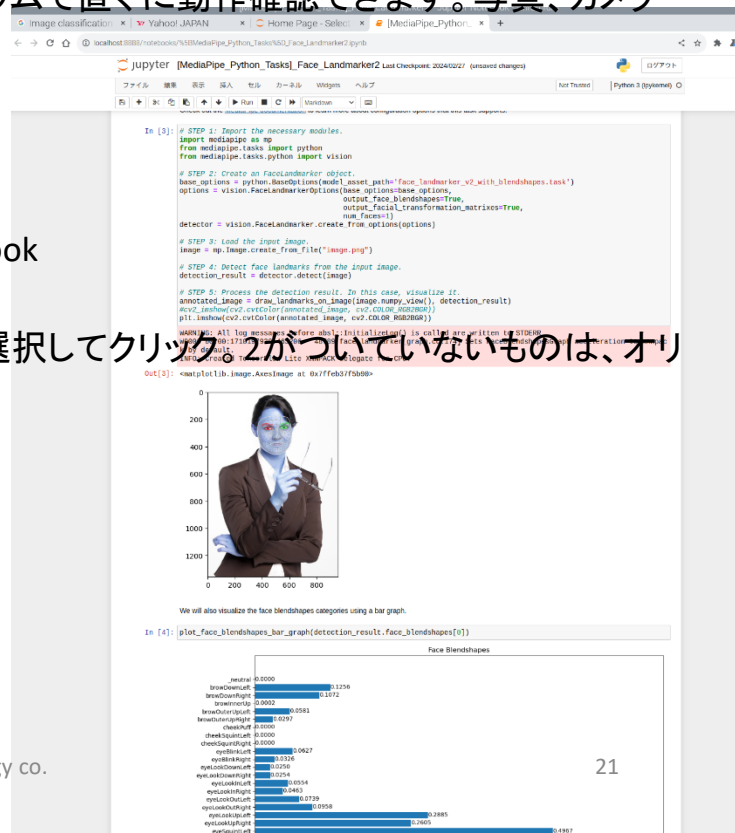
```
$ jupyter notebook
```

[MediaPipe_Python_Tasks]_Face_Landmarker2.ipynb を選択してクリック 2がいないものは、オリジナル

vnc接続では、jupyter notebookは遅い。HDMIで実施推奨

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
$ cd
/home/pi/Documents/mediapipe/mediapipe/notebook
$ jupyter notebook
[MediaPipe_Python_Tasks]_Face_Landmarker2.ipynb
```



1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

C) python:画像

使用方法

仮想化で使用する

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker.py
```

Jupyterと同様に顔のランドマーク検出を行います。

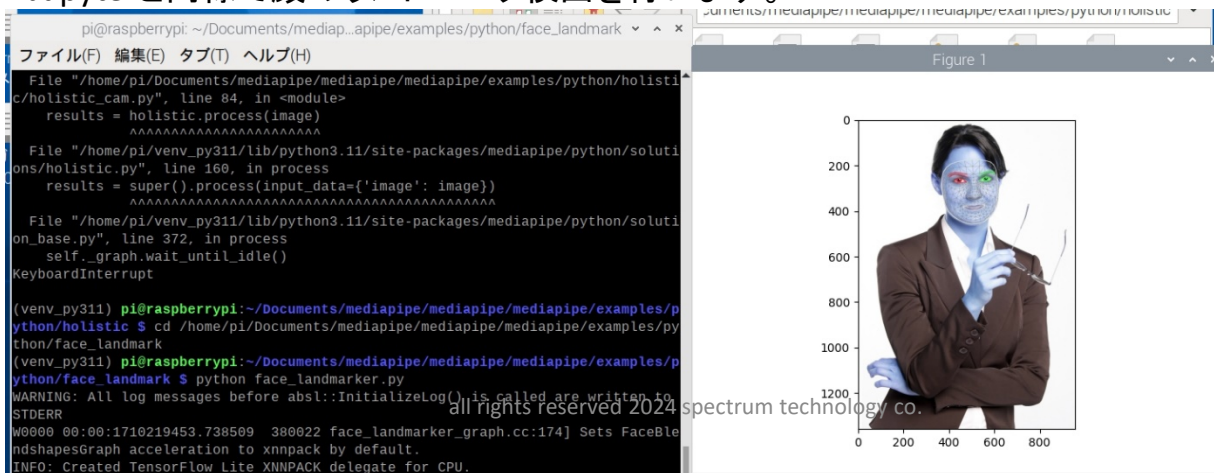
コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd
```

```
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker.py
```



1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
 - B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
 - C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
 - D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ
- C) python:カメラ

使用方法

仮想化で使用

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker_cam2.py
```

顔のランドマークがでます。

fps測定は、

```
$ /home/pi/Documents/mediapipe/mediapipe_sample/examples/face_landmarker/raspberry_pi
```

```
$ python detect.py
```

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
```

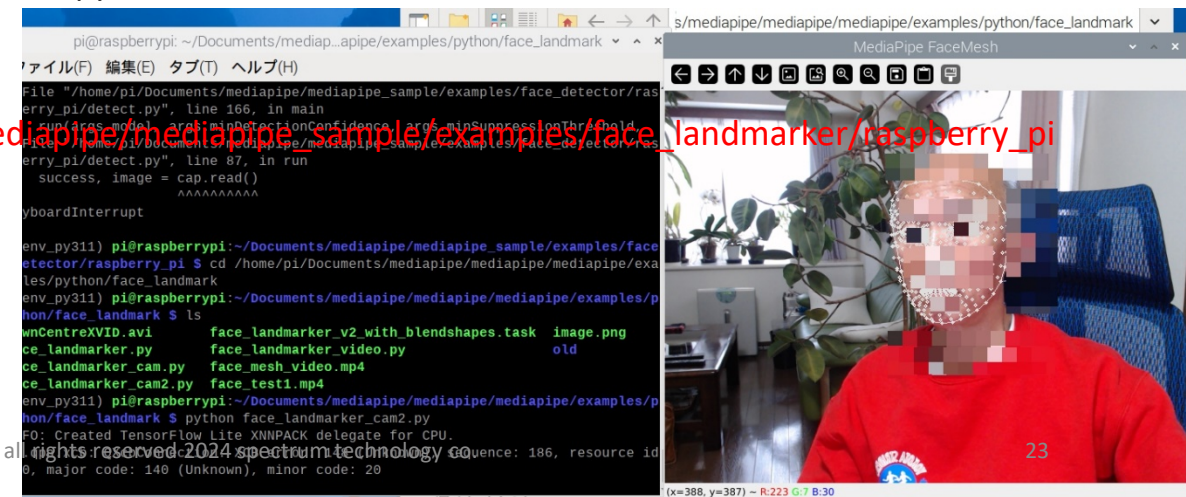
```
$ cd
```

```
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker_cam2.py
```

```
$ /home/pi/Documents/mediapipe/mediapipe_sample/examples/face_landmarker/raspberry_pi
```

```
$ python detect.py
```



1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

C) python:ビデオ

使用方法

仮想化で使用する

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker_video.py --input face_test1.mp4
```

Inputない場合はカメラ

顔のランドマークがでます。

高速

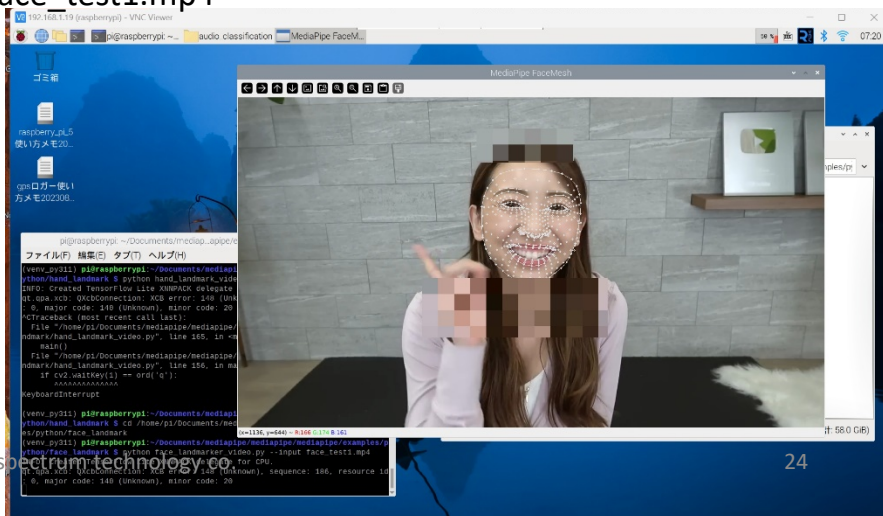
コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd
```

```
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark
```

```
$ python face_landmarker_video.py --input face_test1.mp4
```



1. 画像映像

⑨. 顔ランドマーク検出

画像、映像から顔のランドマークを検出します。

https://developers.google.com/mediapipe/solutions/vision/face_landmarker

方法

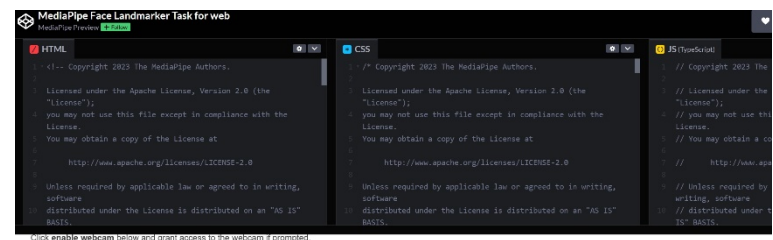
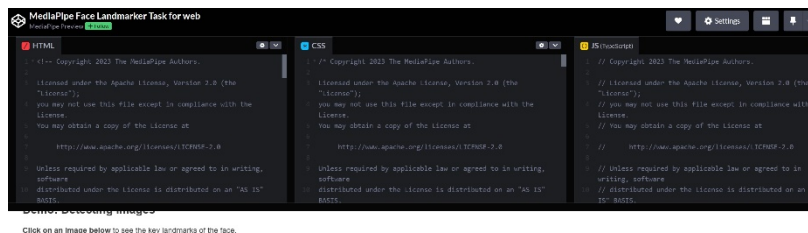
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、写真、ビデオ、カメラで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。写真、カメラ

D) codepen

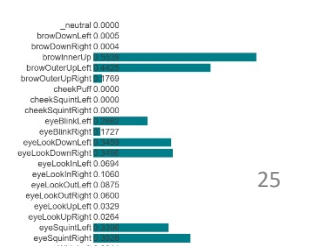
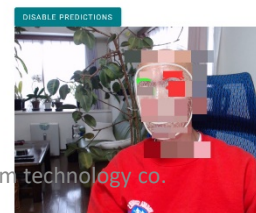
使用方法

<https://codepen.io/mediapipe-preview/pen/OJBVQJm>

- Forkで自分のサイトへ取り込み
- format javascriptを押すと、起動
- 写真のサンプルを押すと、顔のランドマーク検出と確率が出ます。カメラも同様。



all rights reserved 2024 spectrum technology co.



2. テキスト

①. テキスト分類

入力されたテキストをpositive, negativeに分類します。

https://developers.google.com/mediapipe/solutions/text/text_classifier

方法

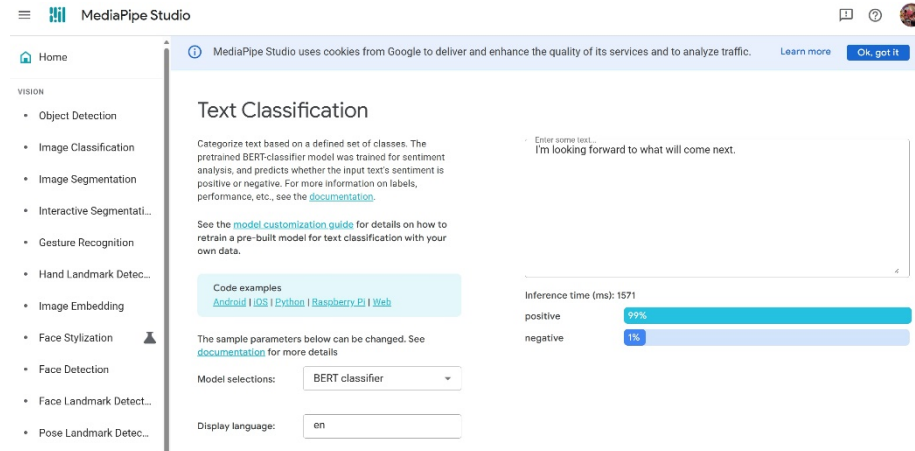
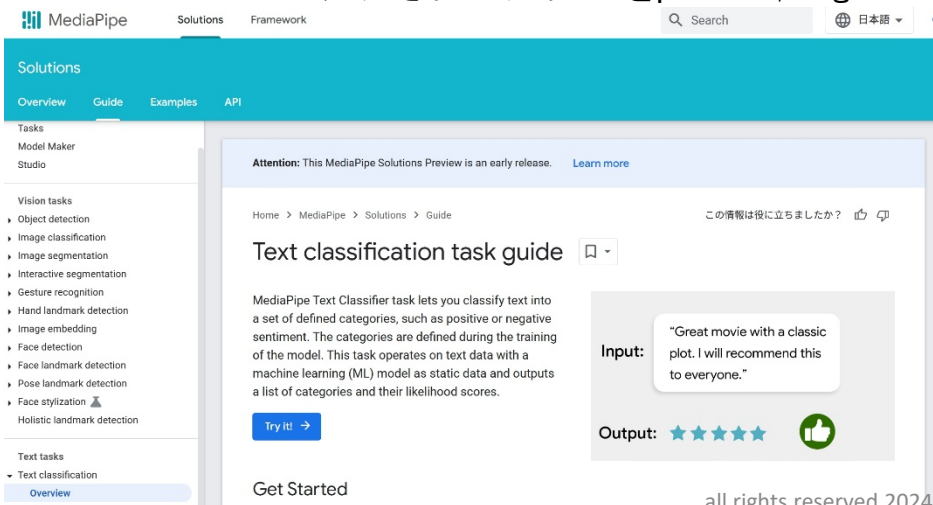
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、テキストで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。テキスト

A) Mediapipe studio

https://mediapipe-studio.webapps.google.com/demo/text_classifier

使用方法

- Mediapipeのサイトからtry itを押すと別画面に起動
- 入力されたテキストをpositive, negativeに分類します。



2. テキスト

①. テキスト分類

入力されたテキストをpositive, negativeに分類します。

https://developers.google.com/mediapipe/solutions/text/text_classifier

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、テキストで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。テキスト

C) python:テキスト

使用方法

仮想化で使用する

```
$ source /home/pi/venv_py311/bin/activate
```

```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/text_classification
```

```
$ python text_classifier.py
```

Jupyterと同様で入力されたテキストをpositive, negativeに分類を行います。

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate  
$ cd  
/home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/text_classification  
$ python text_classifier.py
```

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
first_manager.window.mainloop()  
File "/usr/lib/python3.11/tkinter/__init__.py", line 1485, in mainloop  
self.tk.mainloop(n)  
KeyboardInterrupt  
  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark $  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/face_landmark $ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/pose_landmark  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/pose_landmark $ python pose_landmarker.py  
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.  
Clipping input data to the valid range for imshow with RGB data [[0..1] for floats or [0..255] for integers).  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/pose_landmark $ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/text_classification  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/text_classification $ python text_classifier.py  
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.  
positive (0.99)  
(venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/python/text_classification $
```

2. テキスト

①. テキスト分類

入力されたテキストをpositive, negativeに分類します。

https://developers.google.com/mediapipe/solutions/text/text_classifier

方法

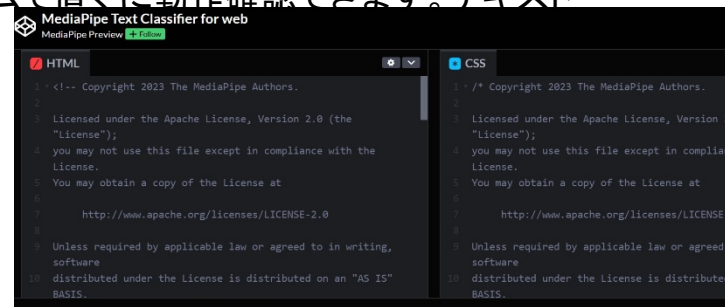
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、テキストで確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。テキスト

D) codepen

使用方法

<https://codepen.io/mediapipe-preview/pen/wvXebMW>

- Forkで自分のサイトへ取り込み
- format javascriptを押すと、起動
- 入力されたテキストをpositive, negativeに分類。



Classifying text with the MediaPipe Text Classifier Task

This demo runs sentiment analysis on the input text. The result shows how likely the input text is to have a positive or negative sentiment.

How to use

Add text to the input field, and then press **Classify** to view sentiment classification.

You can **POPULATE TEXT** with a default input, or add your own text.

Input:

I'm looking forward to what will come next.

CLASSIFY

Sentiment:

positive: 0.99

negative: 0.01

3. 音声

①. 音声分類

音声から話し声、TV、音楽、環境音などをリアルタイムで分類します。

https://developers.google.com/mediapipe/solutions/audio/audio_classifier

方法

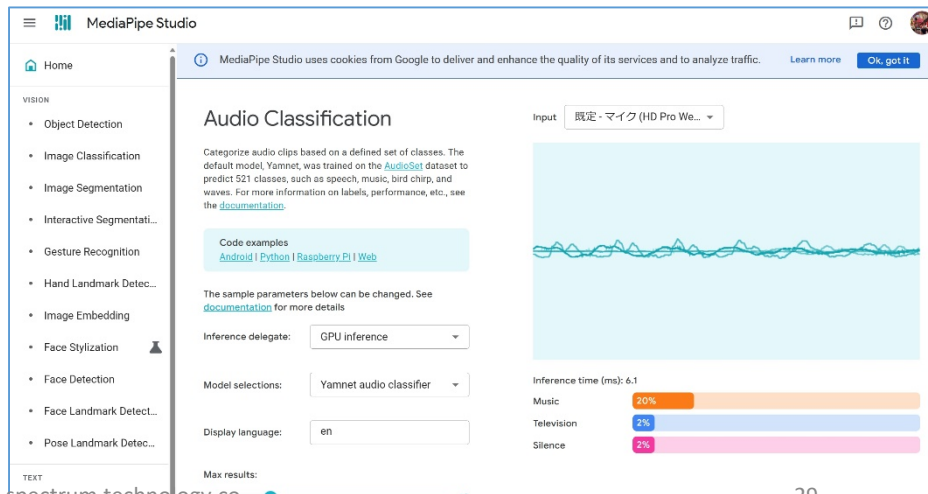
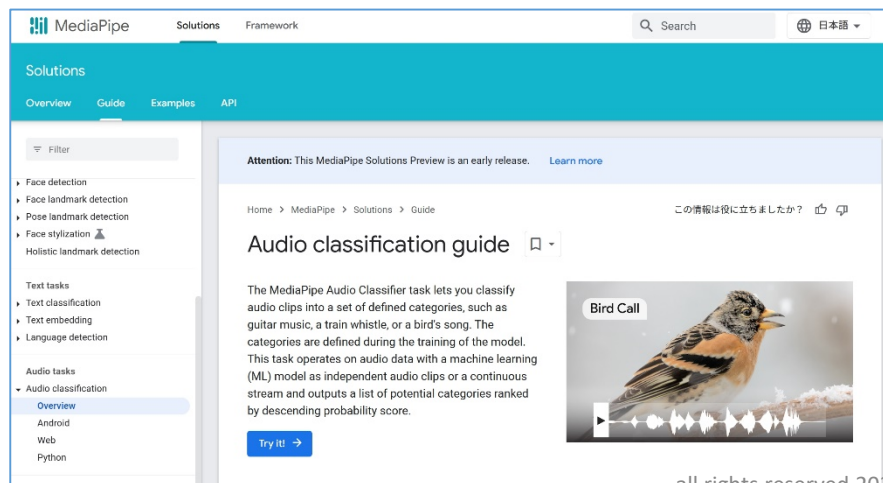
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、音声を確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。音声

A) Mediapipe studio

https://mediapipe-studio.webapps.google.com/demo/audio_classifier

使用方法

- Mediapipeのサイトからtry itを押すと別画面に起動
- 入力された音声から話し声、TV、音楽、環境音など分類します。



3. 音声

①. 音声分類

音声から話し声、TV、音楽、環境音などをリアルタイムで分類します。

https://developers.google.com/mediapipe/solutions/audio/audio_classifier

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、音声を確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。音声

B) Jupyter notebook

使用方法

仮想化で使用

\$ source /home/pi/venv_py311/bin/activate

\$ cd /home/pi/Documents/mediapipe/mediapipe/notebook

\$ jupyter notebook

audio_classification2.ipynb を選択してクリック。2がついていないものは、オリジナル

vnc接続では、jupyter notebookは遅い。HDMIで実施推奨

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate
$ cd
/home/pi/Documents/mediapipe/mediapipe/notebook
$ jupyter notebook
audio_classification2.ipynb
```

```
audio_classification2 Last Checkpoint: 2024/02/26 (unsaved changes)

The first thing you will need to do is install the necessary dependencies for this sample.

In [1]: !pip install -q sounddevice==0.4.4
!pip install -q mediapipe

The next step you will take is downloading an off-the-shelf model for audio classification. In this case you will use the YAMNet model, which is designed to classify audio in 0.975 second segments, though you are also able to use others, including your own custom models, with MediaPipe Tasks.

In [2]: !wget -O classifier.tflite -q https://storage.googleapis.com/mediapipe-models/audio_classifier/yamnet/Float32/1/yamnet.tflite

Performing Audio Classification

Now that you have the necessary dependencies, it's time to start classifying some audio! While there are a variety of ways to retrieve audio clips, this example will download a .wav file of someone speaking.

In [3]: import urllib
audio_file_name = 'speech_16000_hz_mono.wav'
url = 'https://storage.googleapis.com/mediapipe-assets/audio_file_name'
urllib.request.urlretrieve(url, audio_file_name)

Out[3]: ('speech_16000_hz_mono.wav', <http.client.HTTPMessage at 0x77f6a37b0d8>)

You can't show the file directly, but you can download it correctly by displaying a playback widget.

In [4]: from IPython.display import Audio
file_name = 'speech_16000_hz_mono.wav'
display(Audio(file_name, autoplay=False))

Once everything looks good, you can start performing inference. You will start by creating the options that are necessary for associating your model with the Audio Classifier, as well as some other customizations.

Next, you will create your Classifier and read some information from your downloaded audio file, as well as segment the clip into smaller (0.975 seconds, in this case) clips before classifying them.

Finally, you will loop through the audio file in increments of 975 (the amount of seconds per clip in milliseconds) to display the classification results.

In [5]: import numpy as np
from mediapipe.tasks import python
from mediapipe.tasks.python.components import containers
from mediapipe.tasks.python import audio
from scipy.io import wavfile

# Create and associate model for classifier
base_options = python.BaseOptions(model_path='classifier.tflite')
options = audio.AudioClassifierOptions(
    base_options=base_options, max_results=1)

# Create classifier, segment audio clips, and classify
with audio.AudioClassifier.create_from_options(options) as classifier:
    sample_rate, wav_data = wavfile.read(audio_file_name)
    audio_clip = containers.AudioData.create_from_array(
        wav_data.astype(float) / np.iinfo(np.int16).max, sample_rate)
    classification_result_list = classifier.classify(audio_clip)
    assert(len(classification_result_list) == 5)

# Iterate through clips to display classifications
for idx, timestamp in enumerate([0, 975, 1950, 2925]):
    classification_result = classification_result_list[idx]
    top_category = classification_result.classifications[0].categories[0]
    print(f'[{timestamp}] {top_category.category_name} ({top_category.score:.2f})')

Timestamp 0: Speech (0.97)
Timestamp 975: Speech (0.99)
Timestamp 1950: Speech (0.98)
Timestamp 2925: Speech (1.00)

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

3. 音声

①. 音声分類

音声から話し声、TV、音楽、環境音などをリアルタイムで分類します。

https://developers.google.com/mediapipe/solutions/audio/audio_classifier

方法

- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、音声を確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。音声

C) python:テキスト

使用方法

仮想化で使用

```
$ source /home/pi/venv_py311/bin/activate
```

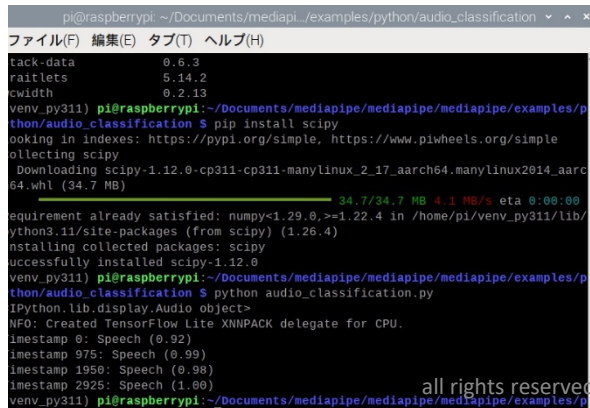
```
$ cd /home/pi/Documents/mediapipe/mediapipe/mediapipe/examples/python/audio_classification
```

```
$ python audio_classification.py
```

Jupyterと同様で入力された音声から話し声、TV、音楽、環境音など分類します。

コマンド入力

```
$ source /home/pi/venv_py311/bin/activate  
$ cd  
/home/pi/Documents/mediapipe/mediapipe/m  
mediapipe/examples/python/audio_classification  
$ python audio_classification.py
```



```
pi@raspberrypi: ~/Documents/mediapipe/examples/python/audio_classification
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
track-data      0.6.3
raitlets        5.14.2
cwidth          0.2.13
venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/examples/p
thon/audio_classification $ pip install scipy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting scipy
  Downloading scipy-1.12.0-cp311-cp311-manylinux_2_17_aarch64.manylinux2014_aarc
64.whl (34.7 MB)
    requirement already satisfied: numpy<1.29.0,>=1.22.4 in /home/pi/venv_py311/lib/
python3.11/site-packages (from scipy) (1.26.4)
Installing collected packages: scipy
Successfully installed scipy-1.12.0
venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/p
thon/audio_classification $ python audio_classification.py
Python.lib.display.Audio object>
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
timestamp 0: Speech (0.92)
timestamp 975: Speech (0.99)
timestamp 1950: Speech (0.98)
timestamp 2925: Speech (1.00)
venv_py311) pi@raspberrypi:~/Documents/mediapipe/mediapipe/mediapipe/examples/p
```

3. 音声

①. 音声分類

音声から話し声、TV、音楽、環境音などをリアルタイムで分類します。

https://developers.google.com/mediapipe/solutions/audio/audio_classifier

方法

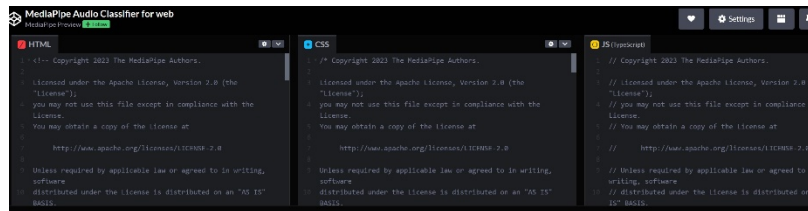
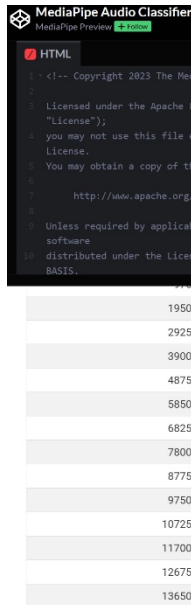
- A) Mediapipe studio:webのデモ版で直ぐに体験できます。
- B) Jupyter notebook: pythonのプログラムで、順番に内容確認します。
- C) Python ローカル: 端末からpythonのプログラムを動作させ、音声を確認できます。
- D) Web:codepen: html,css,javascriptで記述したプログラムで直ぐに動作確認できます。音声

D) codepen

使用方法

<https://codepen.io/mediapipe-preview/pen/wvxYYmy>

- Forkで自分のサイトへ取り込み
- format javascriptを押すと、起動
- 入力された音声から話し声、TV、音楽、環境音など分類します。右はマイクでの実例



Stream audio classification

Click **Start Classifying** to start streaming audio classification.

START CLASSIFYING

Music(0.148)
Silence(0.082)
Speech(0.008)

参考

スマホ(Android, ios)開発

• 物体検出例

- https://developers.google.com/mediapipe/solutions/vision/object_detector

- Android

- Code sample参照

\$ cd

/home/pi/Documents/mediapipe/mediapipe_sample/examples/object_detection/android

- 詳細は、ホームページ参照のこと

- ios

- 項目参照

\$ cd /home/pi/Documents/mediapipe/mediapipe_sample/examples/object_detection/ios

- 詳細は、ホームページ参照のこと

