

# 「NVIDIA・Jetson nanoによるリアルタイム人物カウントについて」

話せばわかるコンピュータの会  
千葉 忠悦

2020年02月22日

1. はじめに

2. Jetson nano導入手順

3. Jetson nano人物カウント実装

4. データ連携 & データ活用

# 1. はじめに

1. はじめに

2. Jetson nano 導入手順

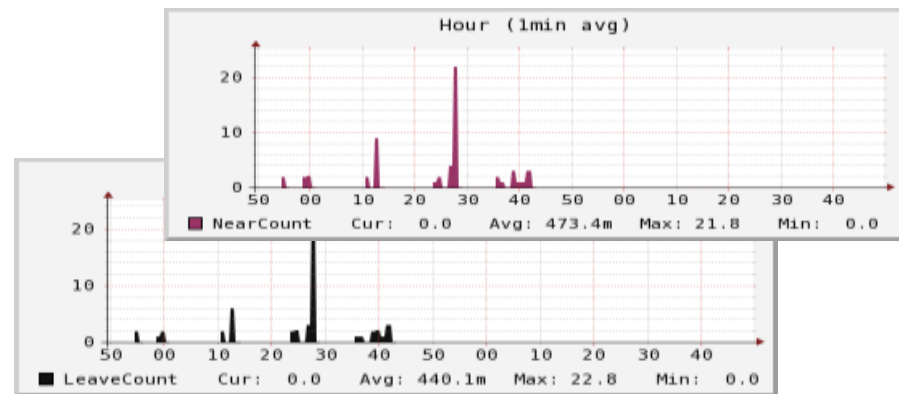
3. Jetson nano 人物カウント実装

4. データ連携 & データ活用

なぜ Jetson nanoを使おうと考えたのか

# お近づきカウンター・ドップラーセンサー版

- 1) ドップラセンサーで通行量をカウント
  - 2) 時系列での通行量の把握
- \*これは、すでに実現していた



近接 ; 2  
離反 ; 2  
リセット

ZigBee

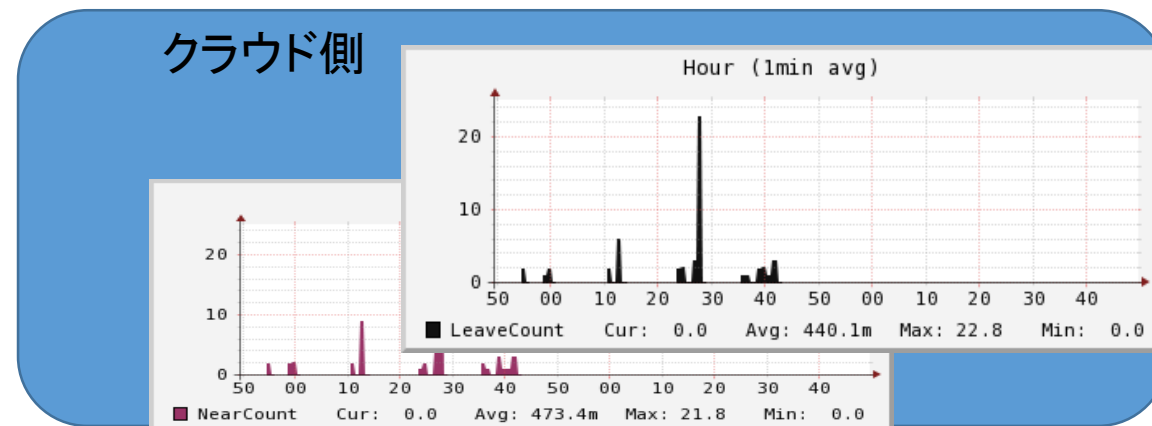


センサーサーバ



# お近づきカウンター・Jetson版で実現したいこと

- 1) 現在ドップラセンサーで実現しているで  
通行量をカウントをJetson Nano+USBカメラで実現する
- 2) 画像認識(deeplearning)で人間の顔が一定時間内に  
存在したらカウントする、同じ顔が画面に居続けたら  
それは1人としてカウントする
- 3) カウント値は現在のドップラセンサーと同様に  
クラウド側に転送する
- 4) クラウド側で時系列での通行量として把握（既存と同様）



近接;2

離反;2

近接、離反判定が難しい場合は  
画面に現れた人をカウントする

USBカメラ例  
BUFFALO 200万画素WEBカメラ 広角120°  
マイク内蔵 ブラック BSW200MBK



NVIDIA Jetson Nano 開発キット



# お近づきカウンター・Jetson Nano版

非構造化データ

構造化データ

データレイク

動画

ディープラーニング

データ可視化

```
{"Date":"11/19 21:26:34","nearCount":"82","LeaveCount":"82"}  
{"Date":"11/19 21:27:34","nearCount":"9","LeaveCount":"9"}
```

カメラ

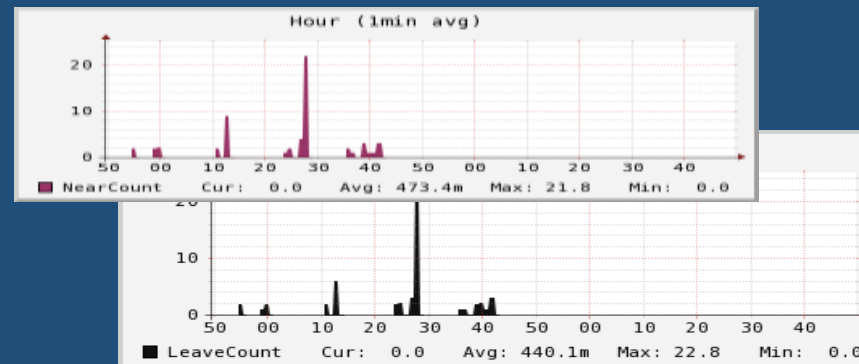


Jetson Nano



近接;2  
離反;2

クラウド側



## ■ 特長

- 1) カメラ画像をリアルタイム解析(ディープラーニング)しPerson 検知
- 2) 単位時間(1分)にカメラに現れた人をカウント
- 3) 機械学習で構造化データに変換

# エッジコンピューティングの立ち位置

①自己完結形 ; エッジ側で全て n 処理を行う

②構造化データ連携型 ; エッジ側で構造化データまで行う

③収集データ連携型 ; エッジ側で取得したデータをすべてクラウド側に連携

④エッジ相互連携型 ; エッジとおしでデータ連携を行う



# エッジコンピューティングの立ち位置

①エッジコンピューティングの性能がUP

②オープンソース・ソフトウェアが充実してきた

③セキュリティ上の問題で、エッジ側で判断させたい事象が増えた



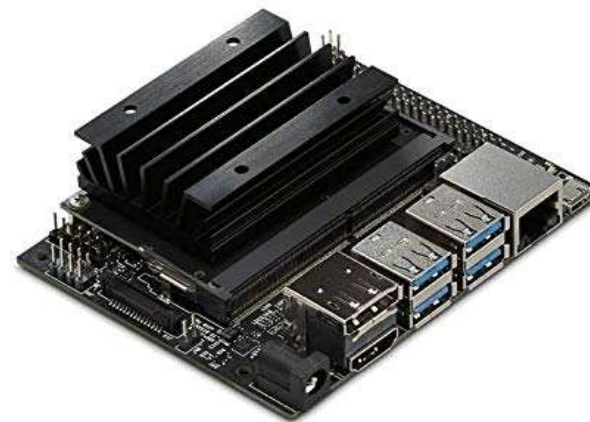


# エッジコンピューティング・活用例

①自走型ロボット

②自走追尾ドローン

③AIスピーカー



## 2. Jetson nano 導入手順

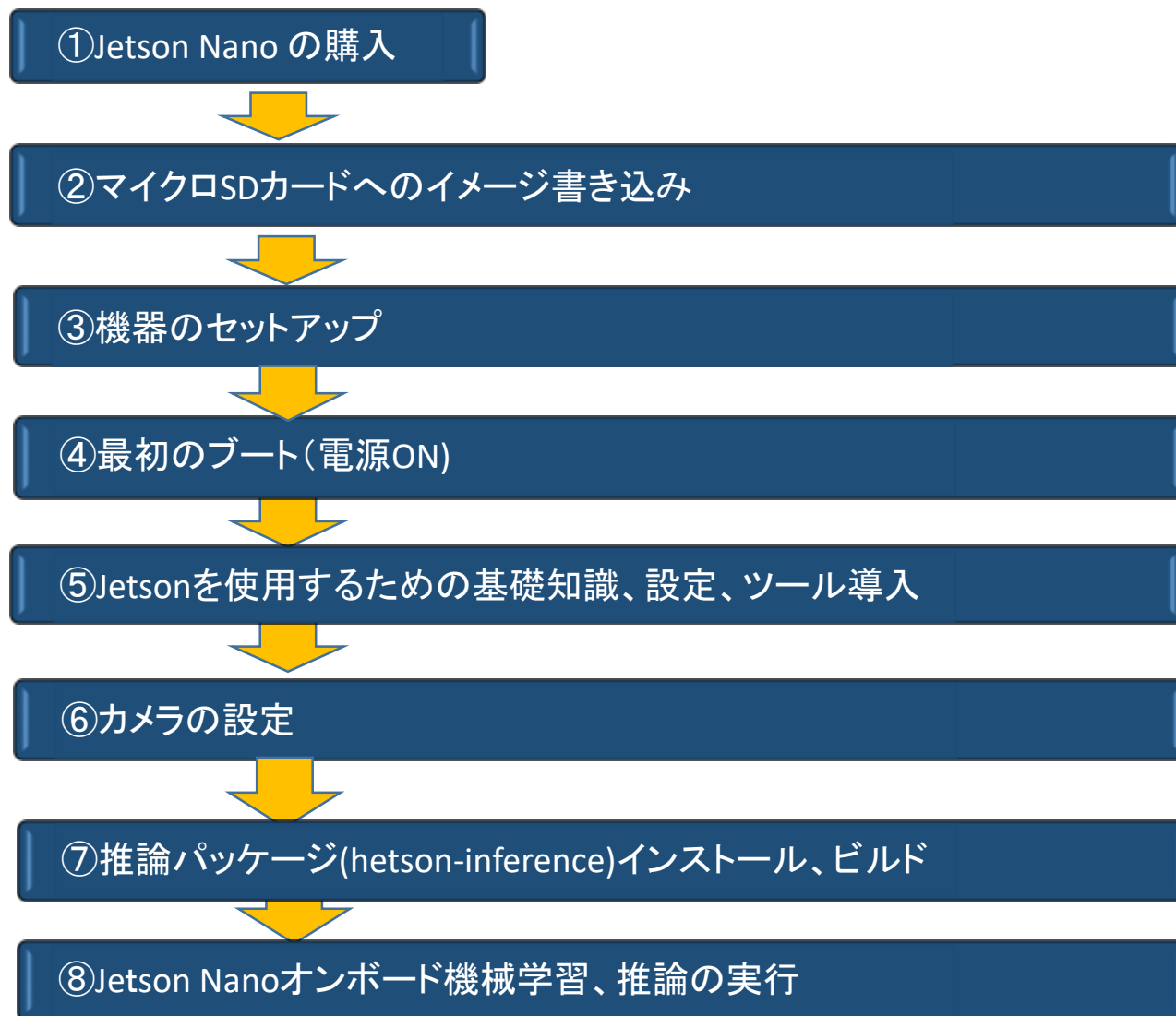
1. はじめに

2. Jetson nano 導入手順

3. Jetson nano人物カウント実装

4. データ連携 & データ活用

# Jetson Nano 導入手順



## ①Jetson Nano 関連部品(価格は目安)

No	項目	仕様	金額(円)
1	Jetson Nano本体	Jetson Nano Developer Kit	13800
2	MicroSDカード	64GB	1000
3	液晶モニター	7インチHDMI LCD 1024x600	8960
4	HDMIケーブル		1000
5	キーボード	英語キーボード	1500
6	マウス	USBマウス	1000
7	カメラ	MX219-D77 モジュール RaspberryPiカメラボードV2	3980
8	ファン、固定アクリル板	Jetson Nano用ファン&ベースプレート	1800
9	電源	5V4.0A	1500
10	Wi-Fi	TP-Link WIFI無線子機11n/11g/11b TL-WN725N	790
		合計	35330



## ①Jetson Nano (SPEC)

	項目	
1	GPU	128 コア Maxwell
2	CPU	クアッドコア ARM A57 @ 1.43 GHz
3	メモリ	4 GB 64 ビットLPDDR4 25.6 GB/秒
4	ストレージ	microSD (別売)
5	ビデオ エンコード	4K @ 30   4x 1080p @ 30   9x 720p @ 30 (H.264/H.265)
6	ビデオ デコード	4K @ 60   2x 4K @ 30   8x 1080p @ 30   18x 720p @ 30   (H.264/H.265)
7	カメラ	2x MIPI CSI-2 DPHY レーン
8	コネクティビティ	ギガビット イーサネット、M.2 Key E
9	ディスプレイ	HDMI および DP
10	USB	4x USB 3.0、USB 2.0 Micro-B
11	その他	GPIO、I2C、I2S、SPI、UART
12	構造	100 mm x 80 mm x 29 mm



## ②マイクロSDカードへのイメージ書き込み

1) Jetson Nano Developer Kit用 SDカードイメージのダウンロード

<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write-windows> これは5. 2GB程度で数時間かかります。

2) SDカードをリードライトできるPCを用意

3) SDメモ리카ードフォーマッターのダウンロード

以下から、ダウンロード、インストールします。

[https://www.sdcard.org/jp/downloads/formatter/eula\\_windows/index.html](https://www.sdcard.org/jp/downloads/formatter/eula_windows/index.html)

SDCardFormattrv5\_WinEn.zipをダブルクリックしインストール

4) SDカードをPCスロットに挿入

5) SDカードフォーマッターを起動し、フォーマット



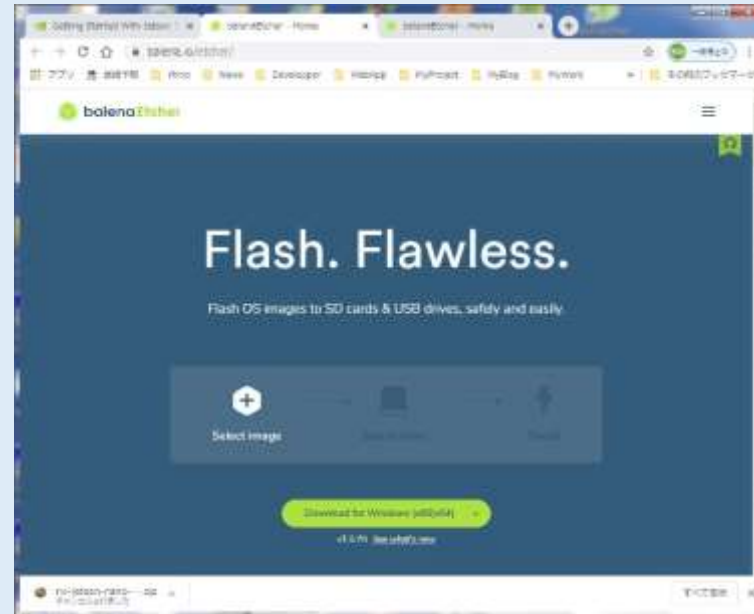


## ②マイクロSDカードへのイメージ書き込み

### 6)SDカードへ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/>



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マイクロSDカードへのイメージ書き込み

2. balenaEtcher を起動します。



3. 「Select image」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ォーマットした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。



### ③機器のセットアップ

- 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、六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



### ③機器のセットアップ

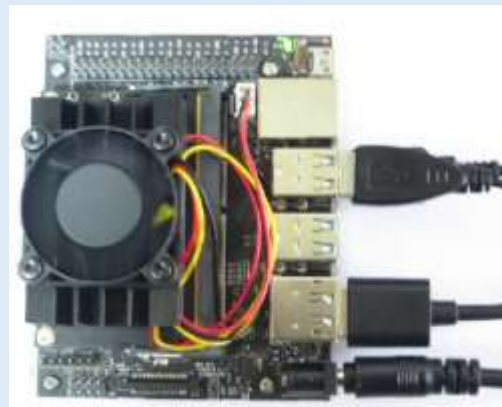
- 6) 電源ジャンパーピンの設定をします。DC ジャックから電源を供給する場合は、J48のヘッダーをジャンパーピンでショートします。USBから供給する場合はオープンにします。



- 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SDカードを挿入します。



- 8) 液晶モニターに電源を入れ、HDMI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ON)

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LEDが点灯します。またファインも回転を始めます。

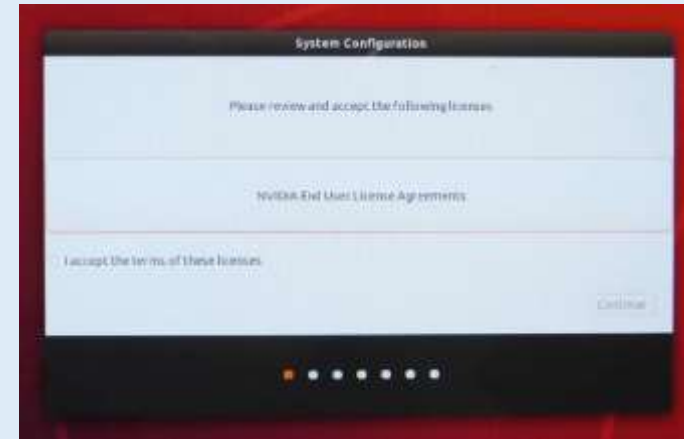
初めてのブート(電源オン)であれば、以下の初期設定が要求されます。

NVIDIA Jetson Nano ソフトウェア使用許諾

EULA(End-User License Agreement)確認

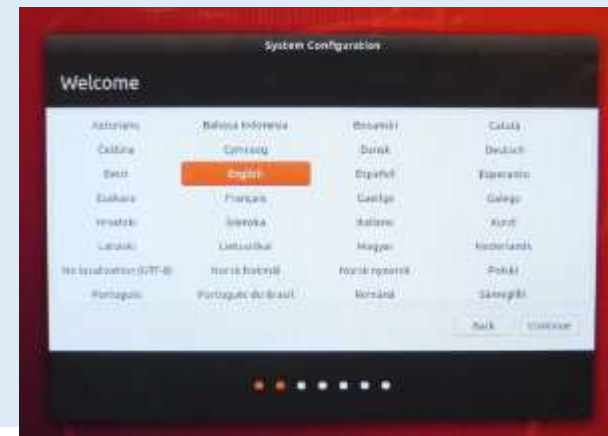
☐ I accept the terms of these licenses

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。



言語選択

English を選択し、[continue]ボタンを押し、次に進みます。

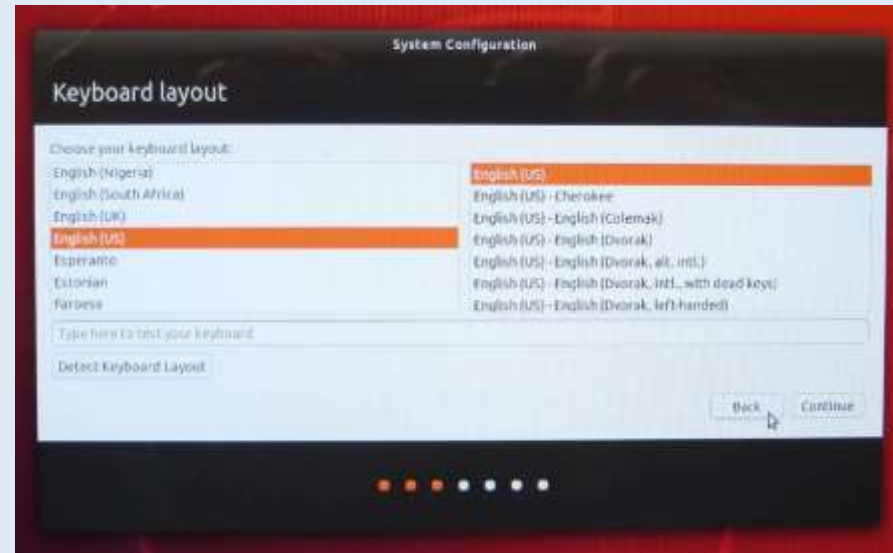




#### ④最初のブート(電源ON)

##### 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。



##### 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。



#### ④最初のブート(電源ON)

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。

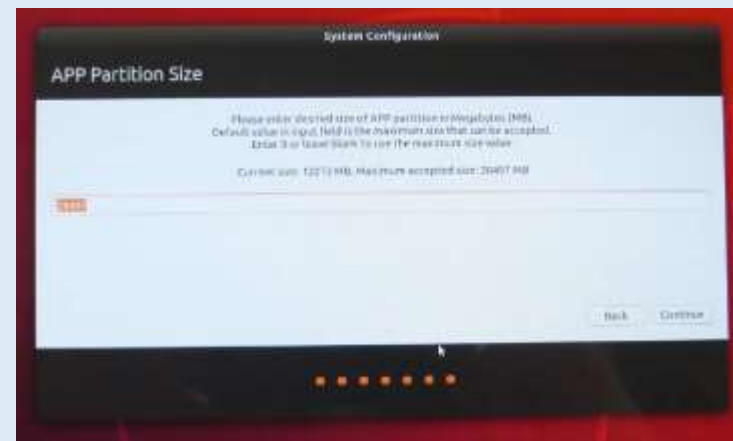
password は6文字以上で入力します。



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。



#### ④最初のブート(電源ON)

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。



## ⑤Jetsonを使用するための基礎知識、設定、ツール導入

- 1) 画面解像度の設定
- 2) 画面ロックのオフ
- 3) 画面スクリーンショットの保存
- 4) 時刻の設定
- 5) ターミナルの起動
- 6) ネットワークの設定
- 7) SSH接続
- 8) FTP接続

## ⑥カメラの設定

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。

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も上げます。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。

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。

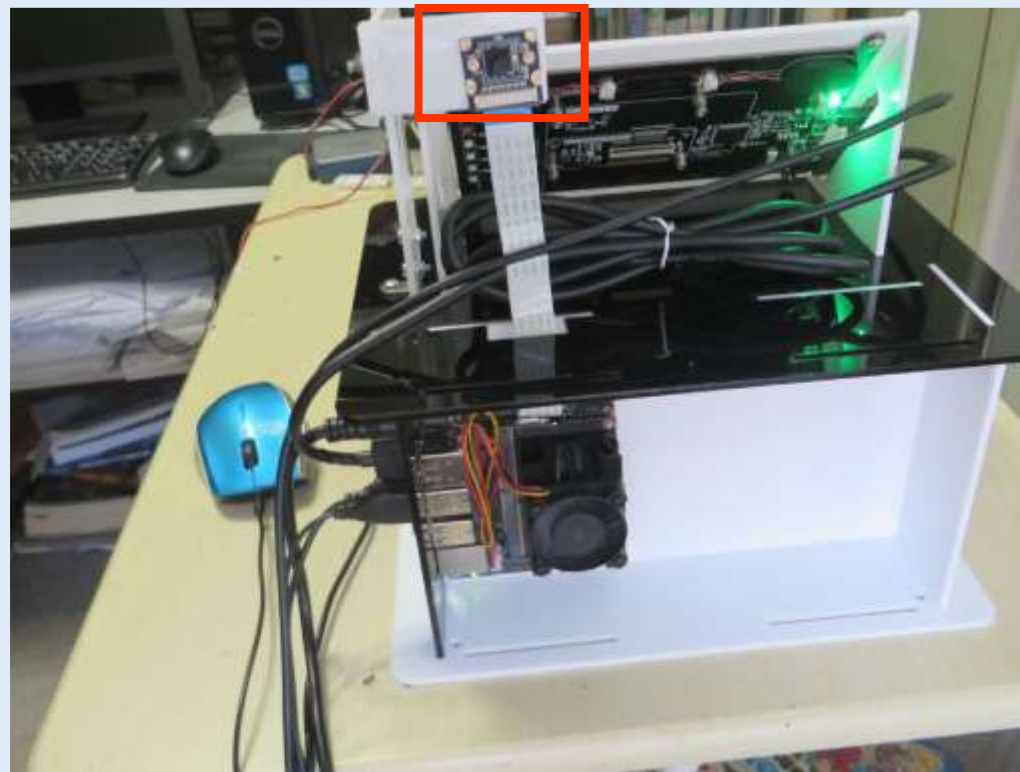
```
#git clone https://github.com/JetsonHacksNano/CSI-Camera
```

## ⑥カメラの設定

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、Simplecamera.py を起動します。

```
#python simplecamera.py
```

以下、カメラ画像が表示されます。





## ⑦推論パッケージ(jetson-inference)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);imageNet

物体検出(object localization);detectNet

セマンティックセグメンテーション(semantic segmentation);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ェア。

CUDA Toolkit (Compute Unified Device Architecture)

## ⑦推論パッケージ(jetson-inference)インストール、ビルド

### JetPackとは

	項目	
1	TensorRT	画像の分類、セグメント化、オブジェクトの検出などの用途のために、リアルタイム・パフォーマンスを最大限に高める <a href="#">ディープラーニング推論エンジン</a> です。
2	cuDNN	畳み込み、アクティベーション機能、テンソル変換などの標準ルーチンに対して、高度に調整された実装を提供するディープラーニング用の <a href="#">CUDA</a> アクセラレーション・ライブラリです。
3	CUDA	<a href="#">GPU</a> 向けの汎用 <a href="#">並列コンピューティング</a> プラットフォーム(並列コンピューティングアーキテクチャ)およびプログラミングモデル
4	マルチメディアAPI	柔軟なアプリケーション開発に最適な低レベルAPIのパッケージ。 カメラAPI: メラのパラメータとEGLストリーム出力に対するフレームごとの制御。 V4L2 API: 動画のデコード、エンコード、フォーマット変換、拡大縮小の機能。
5	VisionWorks	コンピュータビジョン(CV)および画像処理用のソフトウェア開発パッケージ
6	OpenCV	オープンソースのコンピューター・ビジョン・ライブラリです。コンピューターで画像や動画処理するのに必要な、さまざまな機能が実装

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)

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

\* Python またはC++から利用可能

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 SSDMOBILENETV1 91 (COCO classes)

SSD-Mobilenet-v2 ssd-mobilenet-v2 SSDMOBILENETV2 91 (COCO classes)

30

SSD-Inception-v2 ssd-inception-v2 SSDINCEPTIONV2 91 (COCO classes)

DetectNet-COCO-Dog coco-dog COCQDOG dogs

DetectNet-COCO-Bottle coco-bottle COCQBOTTLEbottles

DetectNet-COCO-Chair coco-chair COCQCHAIR chairs

DetectNet-COCO-Airplane coco-airplaneCOCQAIRPLANE airplanes

ped-100 pednet PEDNET pedestrians

multiped-500multipedPEDNETMULTI pedestrians, luggage

facenet-120 facenet FACENET faces

### 3. Jetson nano人物カウント実装

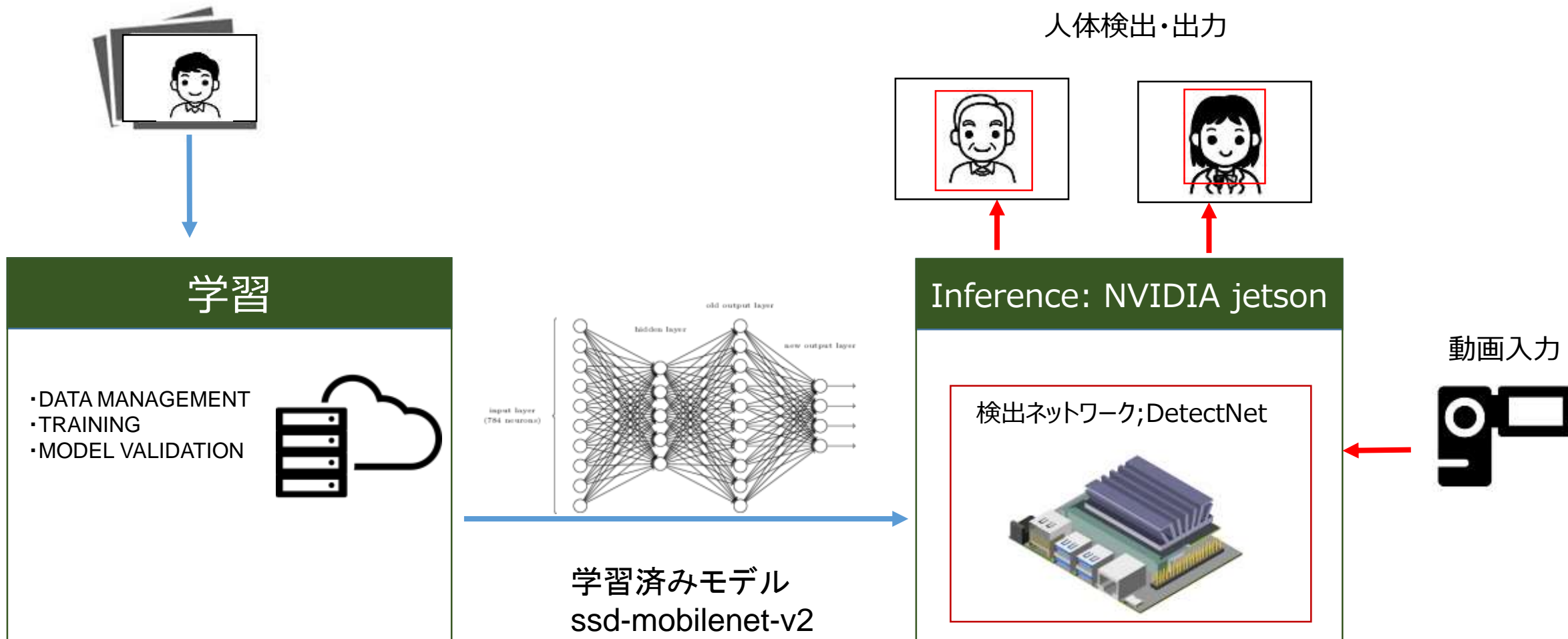
1. はじめに

2. Jetson nano導入手順

3. Jetson nano 人物カウント実装

4. データ連携&データ活用

# 物体検出ネットワーク(DetectNet) 実装イメージ



Mobilenet; 組み込み機器向けの軽量物体認識モデル  
SSD; 物体検出モデル

DetectNet ; NVIDIA の開発した検出ネットワーク

企画

- 何を検出するか
- どの様に検出するか
- どのモデルを使うか
- 検出結果をどうデータ構造化するか

データ準備

- 学習用画像準備
- アノテーション（タグ付け）

学習

- AIモデル構築
- AIモデル検証

予測

- AIモデル実装
- 運用開始
- 構造データのデータ連携

今回は  
学習済みモデル利用



# AIも学習するには時間が必要

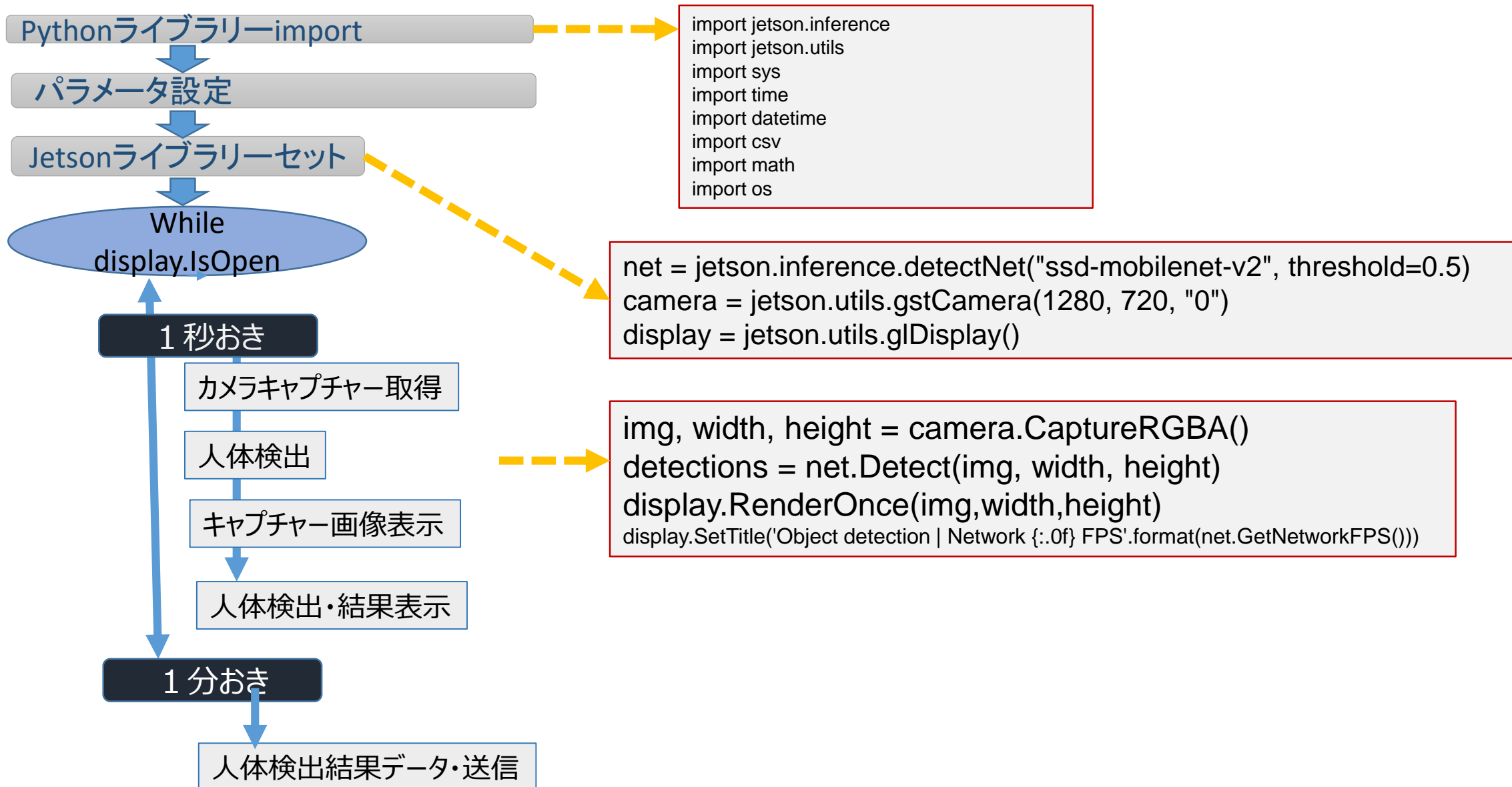
打てども、打てども  
なお、我がゴルフ上手にならざり  
じっと手を見る

なぜ？

3日間全く練習しなかったのに  
なぜか上手くなっている



# 人体検出フロー



## 4. データ連携 & データ活用例

1. はじめに

2. Jetson nano導入手順

3. Jetson nano 人物カウント実装

4. データ連携 & データ活用

# お近づきカウンター・Jetson Nano版

非構造化データ

構造化データ

データレイク

動画

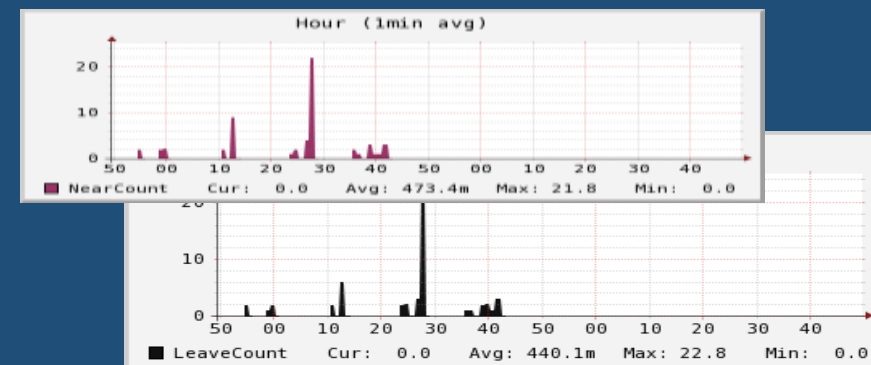
ディープラーニング

データ可視化

```
{"Date": "11/19 21:26:34", "nearCount": "82", "LeaveCount": "82"}  
{"Date": "11/19 21:27:34", "nearCount": "9", "LeaveCount": "9"}
```



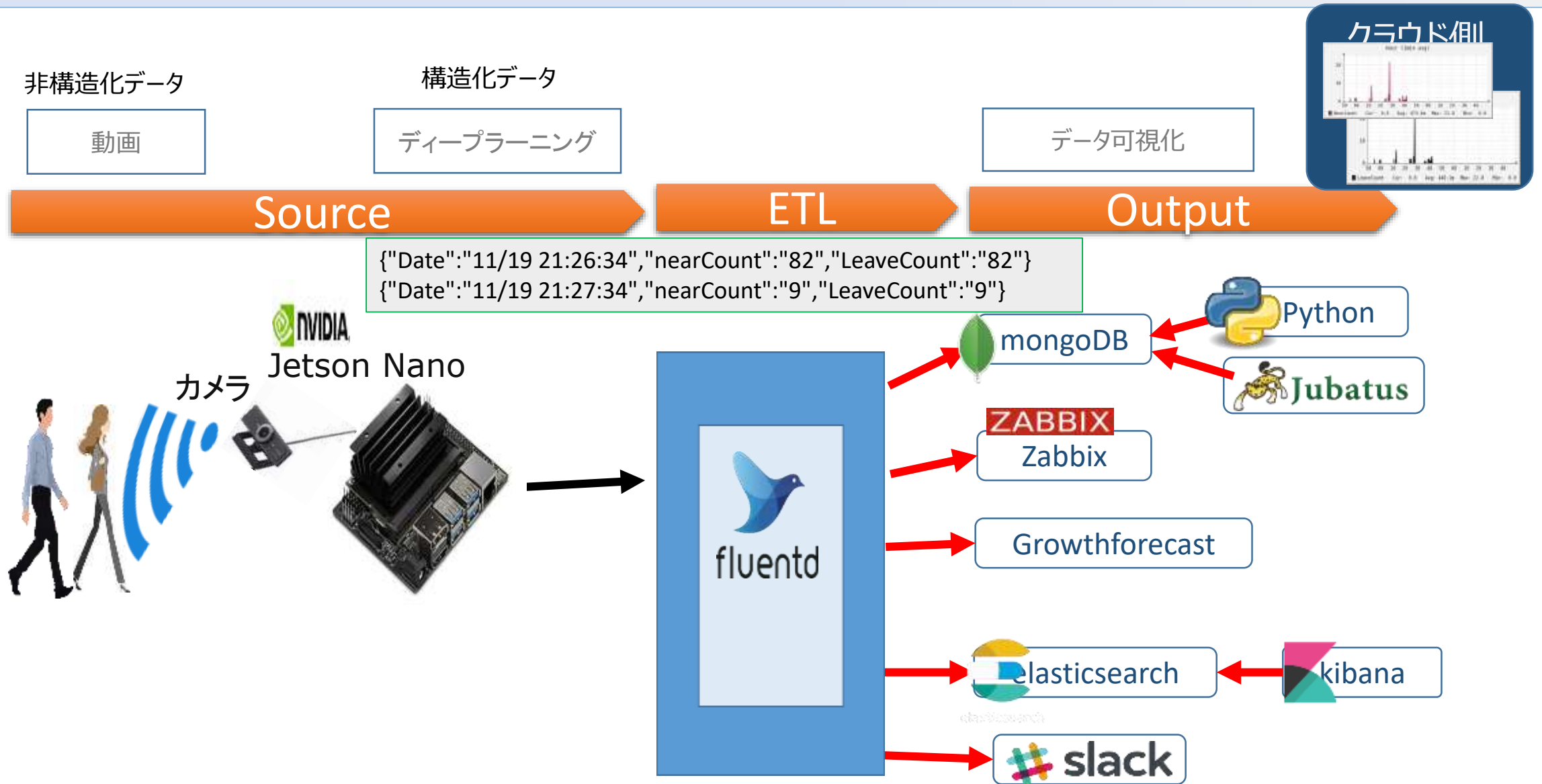
クラウド側



## ■ 特長

- 1) カメラ画像をリアルタイム解析(ディープラーニング)し  
Person 検知
- 2) 単位時間(1分)にカメラに現れた人をカウント
- 3) 機械学習で構造化データに変換

# Fluentdを介したデータ連携図



# UETT

話せばわかるコンピュータの会