Cloud SDK Visualization 機能仕様書

Version 0.1.0 2022 - 11 - 10

Table of Contents (目次)

到新履歴	
はじめに	
]語・略語	3
定ユースケース	5
能概要、アルゴリズム	6
e作性仕様、画面仕様	<u>.</u> 9
l標性能	16
限事項	17
- の他特記事項	18
:決定事項	19

更新履歴

Date	What/Why
2022/11/10	初版作成

はじめに

◆ 本書は、推論結果を確認するためのVisualizationについての機能仕様である。機能開発言語には、TypeScriptを使用する。アプリケーションフレームワークはNext.jsを使用する。

用語・略語

Terms/Abbreviations	Meaning
Console Access Library	Console for AITRIOSへのアクセス方法を提供するSDK
Console for AITRIOS	エッジからクラウドを含めたソリューションを 効率的に導入するための各種機能(デプロイメ ント機能、リトレーニング機能、デバイスマネ ジメント機能など)を提供するクラウドサービ ス
推論結果	Vision and Sensing Applicationからの出力の うち、AI処理されたメタデータ
画像	Vision and Sensing Applicationからの出力の うち、カメラがとらえているイメージデータ

参照資料

- ◆ Visualizationで利用するTypeScript版Console Access Library
 - https://github.com/SonySemiconductorSolutions/aitrios-sdk-console-accesslib-ts
- ◆ Visualizationで利用する推論結果Deserializeサンプルコード
 - https://github.com/SonySemiconductorSolutions/aitrios-sdk-deserialization-sample
- Next.js
 - https://nextjs.org/

想定ユースケース

◆ ユーザーはリポジトリ内のVisualizationを起動することで、エッジAIデバイスがUpload した推論結果の確認ができる。

機能概要、アルゴリズム

Functional Overview

- ◆ Device IDを選択することで、メイン画面が表示される。
- ◆ RealtimeModeとHistoryModeの切り替えができる。
- RealtimeMode
 - ◆ 最新データを取得、表示、保存できる。
 - ◆ Start Uploadボタンを押下することで、画像/推論結果のアップロードを開始する。
 - ◆ Stop Uploadボタンを押下することで、画像/推論結果のアップロードを停止する。
 - ◆ Start Pollingボタンを押下することで、最新の画像/推論結果の取得、保存、画面表示を開始する。
 - ◆ Stop Pollingボタンを押下することで、最新の画像/推論結果の取得、保存、画面表示を停止する。

HistoryMode

- ◆ 指定したサブディレクトリに紐づく過去のデータを取得、表示、保存できる。
- ◆ Start Playingボタンを押下することで、過去の画像/推論結果の取得、保存、画面表示を開始する。
- ◆ Stop Playingボタンを押下することで、過去の画像/推論結果の取得、保存、画面表示を停止する。
- ◆ 任意のLabel名設定用ファイルを設定できる。※Label名設定用ファイルの詳細は制限事項 に記載
- ◆ Docker in Docker で動かすことができる。
- ◆ Codespaces で動かすことができる。
- ◆ ベースAIモデルはObject Detectionのみ対応する。

Algorithm

- 1. 画面を起動する。
 - a. Device IDを入力する。
- 2. メイン画面に移行する。
 - a. モードを選択する。(RealtimeMode/HistoryMode)。
 - b. Label名設定用ファイルを設定する。(指定なしの場合デフォルトのファイルが適用される)

3. RealtimeModeの場合

- a. Probabilityを設定し、Start Uploadを押下する。
- b. startuploadinferenceresultが呼び出され、推論結果と画像のアップロードが開始される。
- c. Polling Intervalを設定し、Start Pollingボタンを押下する。
- d. getImageData,getLatestInferenceDataが定期呼び出しされ、最新の推論結果と画像を取得し、Visualizationが動作しているフォルダに保存する。
- e. 取得した画像と推論結果は画面に表示される。
- f. Stop Uploadボタンを押下すると、stopuploadinferenceresultが呼び出され、推論 結果と画像のアップロードが停止される。
- g. Stop Pollingボタンを押下すると、最新の推論結果と画像の取得、保存、表示が停止される。

4. HistoryModeの場合

- a. History Mode画面に遷移後、Consoleに保存済のDevice IDに紐づく推論結果をすべて取得する。
- b. getImageListを呼び出し、取得したサブディレクトリ一覧からプルダウンを作成する。
- c. Probability/IntervalTime/SubDirectoryを設定し、Start Playingを押下する。
- d. 指定したSubDirectoryを引数に取るgetImageListが呼び出しされ、既にConsoleに格納されている画像を取得する。
- e. 取得した画像と推論結果を、Visualizationが動作しているフォルダに保存し、保存された画像群をもとに、画像名リストを作成する。
 - 保存ディレクトリはRealtimeModeと同様の構成とする。
- f. 画像名リストから取得(保存)された画像の総数を算出し、ImageSelectionを選択できるようにする。
- g. getInferenceDataを呼び出し、取得された画像に紐づく推論結果を取得する。
- h. Localに保存された画像データを取得し、推論結果と合わせて画面に表示される。
- i. Stop Playingボタンを押下すると、推論結果と画像の表示が停止される。

Under what condition

- ◆ Consoleへのアクセスができること。
- ◆ TypeScriptの開発環境が構築されていること。

- ◆ TypeScriptのversionは4.7。
- ◆ そのまま実行する場合に必要である。
- ◆ Dockerが利用可能であること。
 - ◆ Dockerコンテナをビルドして利用する場合に必要である。
- ◆ エッジAIデバイスがConsoleに接続されており、Consoleからの操作を受けつける状態である。

<u>API</u>

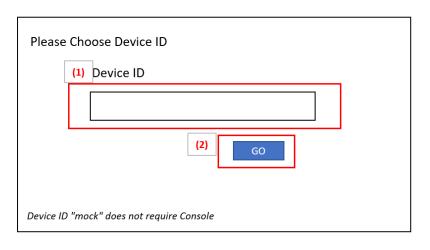
- **♦** GET
 - {base_url}/api/image/imagepath
 - {base_url}/api/imagelist/deviceID?[uploadSubDirectory]
 - ◆ {base_url}/api/inference/deviceID/selector
 - {base_url}/api/labels
 - {base_url}/api/startuploadinferenceresult/deviceID
 - ◆ {base_url}/api/stopuploadinferenceresult/deviceID
 - {base_url}/api/deviceinfo/deviceID

Others Exclusive conditions / specifications

◆ なし

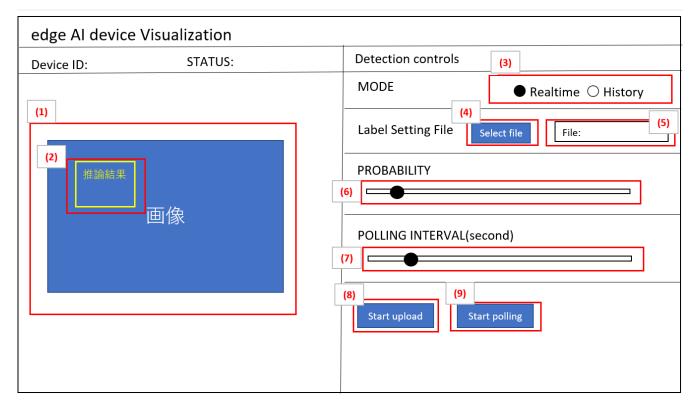
操作性仕様、画面仕様

画面仕様



2 1. Device ID ?????

- 1. 推論結果を確認したい "Device ID" を入力して設定する。
- 2. "Device ID" を確定し結果確認画面に遷移する。
 - a. **"Device ID"** に **"mock"** と入力した場合、Consoleへは接続せず、Visualizationが ダミー動作をする。



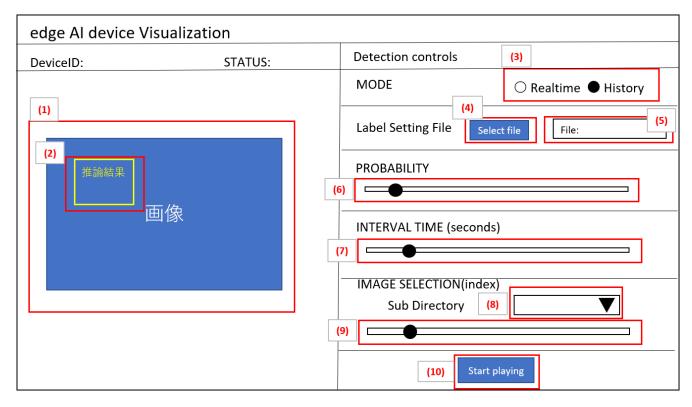
2. RealtimeMode?

STATUSの表示は下記の通りとする。

	Polling中	Polling停止中
Upload中	Uploading / Polling	Uploading / Not polling
Upload停止中	Not Uploading / Polling	Not Uploading / Not polling

- 1. クラウドから取得した最新画像
- 2. クラウドから取得した最新推論結果
- 3. モードを切り替える
- 4. Label名設定用ファイルを選択するボタン(エクスプローラーが表示)
 - ◆ 指定なしの場合は、デフォルトのファイルが適用される src/public/labels/labels.json
- 5. 選択されたFile名を表示する
- 6. 画面上に表示する確信度の境界値を調整する
 - ◆ 境界値はスライドバーの右横に数字で記載
- 7. Consoleヘデータ取得する時のPolling間隔を設定する
 - ◆ Polling間隔はスライドバーの右横に数字で記載
- 8. 推論結果と画像のアップロードを開始/停止するボタン

9. クラウドストレージから最新結果の取得と表示を開始/停止するボタン



2 3. HistoryMode 22

"Device ID" の表示は、"Device ID" 入力画面で入力したDevice IDを表示する。 STATUSの表示は下記の通りとする。

Playing中	Playing停止中
Playing	Not playing

- ◆ 1~5はRealtimeModeの画面と同様
- ◆ 6. 画面上に表示する確信度の境界値を調整する
 - ◆ 境界値はスライドバーの右横に数字で記載
- ◆ 7. 画像を切り替えるPlaying間隔を設定する
 - ◆ Playing間隔はスライドバーの右横に数字で記載
- ◆ 8. クラウド上画像のSubDirectoryをリストから選択する
 - ◆ Consoleに格納されている画像パス一覧からリストを作成
- ◆ 9. 表示開始する画像のインデックスを設定する
 - ◆ インデックスはスライドバーの右横に数字で記載
 - ◆ スライドバーを変更した際は、画像の日時のみインデックスに紐づくものに更新される

◆ 10. 画像切り替えを開始/停止するボタン

操作性仕様

Visualization起動までの操作

- ◆ リポジトリをcloneして使う
 - 開発者は任意のブラウザからVisualizationのリポジトリを開き、リポジトリをCloneする。
 - 2. CloneしたVisualizationに必要なパッケージをインストールする。
 - 3. Visualizationを起動する。
- ◆ Dockerコンテナをビルドして使う
 - 1. 利用者は任意のブラウザからVisualizationのリポジトリを開き、リポジトリをCloneする。
 - 2. リポジトリ内のDockerfileに対し、"下記コマンドを実行することでVisualizationを 起動する。

docker build

Visualization起動後の操作

- 1. **"Device ID"** を入力すると、メイン画面に移行する。
- 2. RealtimeMode/HistoryMode任意のモードを指定する。
- 3. Label名設定用ファイルを選択する(任意)。
 - ◆ 選択されたファイルは、下記に反映される src/public/labels/customLabels.json
 - ◆ 指定なしの場合は、デフォルトのファイルが適用される src/public/labels/labels.json
- 1. RealtimeModeの場合
 - "Probability" / "PollingInterval" を設定し、[Start Upload] 押下後、[Start Polling] ボタンを押下する。
 - ◆ Polling中 "**Probability**" は機能するが、"**PollingInterval**" は機能しない。
 - ◆ 保存ディレクトリは下記の構成とする。

```
public
  LDeviceID
  LYYYYMMDDhhmmddsss ①
  LYYYYMMDDhhmmddsss ②
  LYYYYMMDDhhmmddsss.jpg ③
  LYYYYMMDDhhmmddsss.json ④
```

- ① [StartUpload]開始時に作成されるフォルダ 推論開始時刻がフォルダ名となる
- ② 推論結果と画像を格納するフォルダ 推論結果と画像を保存した時刻がフォルダ名となる
- ③ 画像ファイル
- ④ 推論結果
- 2. HistoryModeの場合

"Probability"/"IntervalTime"/"SubDirectory"/"ImageSelection" を設定し、[Start Playing] を押下する。

- ◆ Playing中 **"Probability"** は機能するが、**"IntervalTime"/"SubDirectory"** /**"ImageSelection"** は機能しない。
- ◆ Stop Playing中に "ImageSelection" を移動させても画像は切り替わらない。
- ◆ 保存ディレクトリはRealtimeModeの場合に記載のものと同様。
- 3. 指定した画像/推論結果の取得を開始し、画面上に表示される。
- 4. RealtimeModeの場合、**[Stop Upload]** ボタン押下後、**[Stop Polling]** ボタンを押下する。
- 5. HistoryModeの場合、[Stop Playing] ボタンを押下する。

各BlockでのAPIパラメータ

GET

◆ {base_url}/api/image/imagepath** 指定したimagepathの画像を取得し返却する。

Query Parameter's name	Meaning	Range of parameter
imagepath	取得したい画像のクラウドスト レージ上のPath	指定なし

Return value	Meaning
buff	取得した画像のバイナリデータ

- ◆ {base_url}/api/imagelist/deviceID?[uploadSubDirectory]
 - ◆ Device IDの画像リストを取得し返却する。

Query Parameter's name	Meaning	Range of parameter
deviceID	画像をUploadしているDevice ID	指定なし
uploadSubDirectory	Console上に保存される画像データフォルダのSubDirectory	指定なし

Return value	Meaning
return	取得した画像リスト

- ◆ {base_url}/api/inference/deviceID?[imagePath]&[timestamp]
 - ◆ 画像に紐づく推論結果リストを取得し返却する。

Query Parameter's name	Meaning	Range of parameter
deviceID	推論結果をUploadしているDevice ID	指定なし
imagePath	推論結果と紐づく保存済み画像 データのpublicフォルダ以下の パス	指定なし
timestamp	推論結果と画像を保存した時刻	指定なし

Return value	Meaning
return	推論結果が含まれたリスト

- ♦ {base_url}/api/labels
 - ◆ 推論結果のClassIDと対応するLabel設定を取得し返却する。

Query Parameter's name	Meaning	Range of parameter
------------------------	---------	--------------------

Return value	Meaning
labels	ClassLabelデータ

- ◆ {base_url}/api/startuploadinferenceresult/deviceID
 - ◆ 指定したDevice IDに対して推論結果と画像のUpload開始を要求する。

Query Parameter's name	Meaning	Range of parameter
deviceID	画像と推論結果をUploadさせるDevice ID	指定なし

Return value	Meaning
result	SUCCESSかERRORの文字列
outputSubDir	画像がアップロードされるディレクトリ

- ◆ {base_url}/api/stopuploadinferenceresult/deviceID
 - ◆ 指定したDevice IDに対して推論結果と画像のUpload停止を要求する。

Body Parameter's name	Meaning	Range of parameter
deviceID	画像と推論結果のUploadを停止させるDevice ID	指定なし

Return value	Meaning
result	SUCCESSかERRORの文字列

- ◆ {base_url}/api/deviceinfo/deviceID
 - ◆ 指定したDevice IDの情報を取得する。

Body Parameter's name	Meaning	Range of parameter
deviceID	取得する対象のDevice ID	指定なし

Return value	Meaning
device information	エッジAIデバイス情報

目標性能

- ◆ UIの応答時間が1.2秒以内であること。
- ◆ 処理に5秒以上かかる場合は、処理中の表現を逐次更新表示できること。

制限事項

- ◆ 今回はObject Detectionのみの対応とする。
- ◆ Label名設定用ファイルはjson形式とする(下記を参考とする)。

```
[
    "Label_1",
    "Label_2",
    "Label_3"
]
```

- ◆ Console UIから、Command Parameter Fileを下記の設定にしておく。
 - ◆ Mode=1(Image&Inference Result)
 - UploadMethodIR="Mqtt"

その他特記事項

◆ エッジAIデバイスからConsoleへの画像アップロード時に、最大数分程度の遅延が発生することがある。

未決定事項

◆ 無し