SONY

Vision and Sensing Application SDK Post Vision App 機能仕様書

Version 0.1.0

2022 - 11 - 10

Table of Contents（目次）

更新履歴

Date	What/Why
2022/11/10	初版作成

用語・略語

Terms/Abbreviations	Meaning
Post Vision App	後処理（AIモデルの出力となるOutput Tensorの加工処理)
PPL	エッジAIデバイスのAIモデルの出力(Output Tensor)を加工処理するモジュール
Wasm	WebAssemblyの略。仮想マシン用のバイナリ命令形式
FlatBuffers	シリアライゼーションライブラリ

参照資料

- Reference/Related documents（関連資料）
  - WebAssembly
    - https://webassembly.org/
  - IMX500用PPLライブラリAPI規格書
    - https://developer.aitrios.sony-semicon.com/development-guides/documents/specifications/
  - FlatBuffers
    - https://google.github.io/flatbuffers/

想定ユースケース

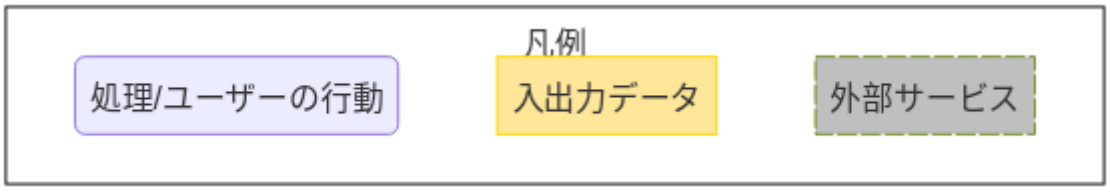
- ◆ 後処理を設計、実装したい
- ◆ 後処理コードを、エッジAIデバイスにデプロイ可能な形式に変換したい

機能概要、アルゴリズム

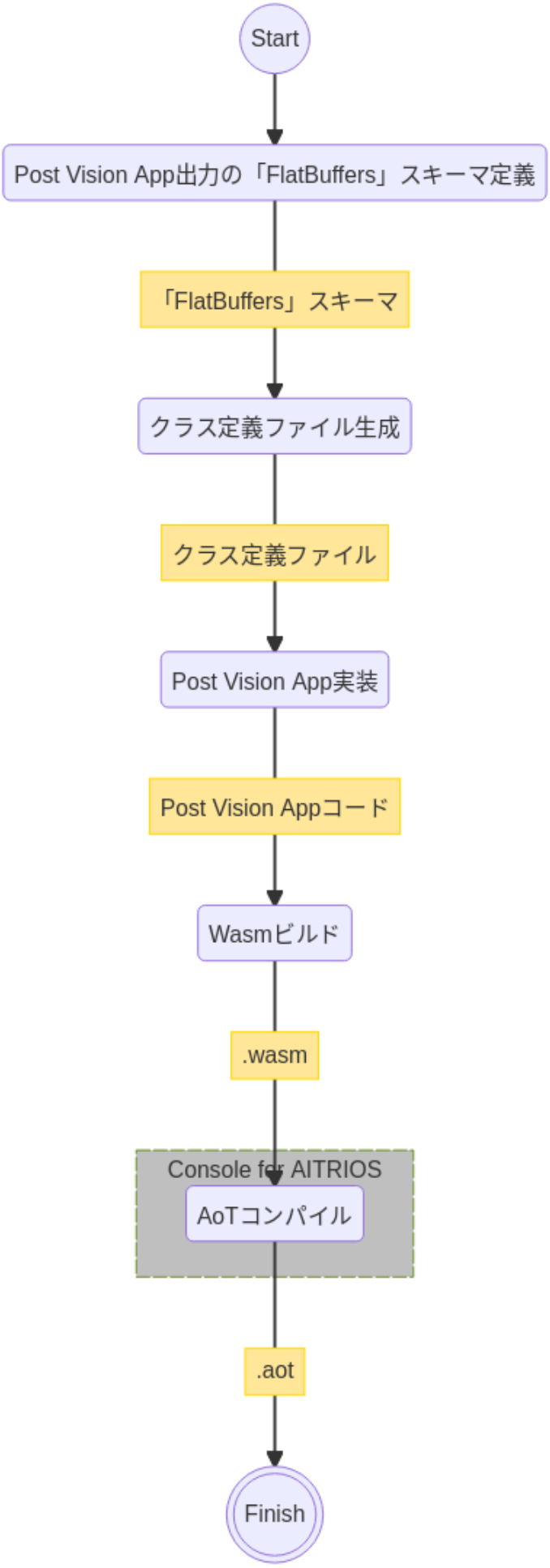
Functional Overview

- C言語、または、C++言語でPost Vision Appを設計、実装できる
- Post Vision Appの出力を「FlatBuffers」でシリアライズできる
  - 「FlatBuffers」のスキーマファイルからクラス定義ファイルを生成できる
- C言語、または、C++言語で実装したPost Vision AppをWasmにビルドできる
- Post Vision Appサンプル環境が提供されている
- Post Vision Appのサンプルコードを使用してWasmビルドができる

Post Vision App作成フロー



フロー



SDK環境で作成したWasmファイルはConsole for AITRIOSでAoTコンパイルを実施し、エッジAIデバイスにデプロイ可能な形式に変換する。

ビルド機能

下記のビルド機能を提供する。

- Wasmビルド
  エッジAIデバイスにデプロイするために、Wasmファイル（.wasm）を生成する。
  - Post Vision Appコード（.c、または、.cpp）からWasmファイル（.wasm）を生成する
    - なお、Wasmビルドの過程で中間生成物としてオブジェクトファイル（.o）を生成する

操作性仕様、画面仕様

How to start each function

1. SDK環境を立ち上げ、Topの **`README.md`** をプレビュー表示する
2. SDK環境Topの **`README.md`** に含まれるハイパーリンクから、 **`tutorials`** ディレクトリの **`README.md`** にジャンプする
3. **`tutorials`** ディレクトリの **`README.md`** に含まれるハイパーリンクから、post process ディレクトリの **`README.md`** にジャンプする
4. post processディレクトリの各ファイルから各機能に遷移する

Post Vision Appの設計・実装

1. **`README.md`** の手順に従って、Post Vision App出力の「FlatBuffers」スキーマファイルを作成する
2. **`README.md`** の手順に従って、VS Code UIからターミナルを開き、スキーマファイルからクラス定義のヘッダファイルを生成するコマンドを実行する
   - スキーマファイルと同階層にクラス定義のヘッダファイルが生成される
3. Post Vision Appの実装を行う
   - 実装はC言語、または、C++言語で行う
   - 実装に使用するソースファイルは新規作成するか、提供しているPost Vision Appのサンプルコードを修正して使用する
   - 「2.」で生成されたクラス定義ファイルを使用して実装を行う
   - 提供しているPost Vision Appのサンプルコードを参考に、Post Vision Appのインタフェースを実装する
   - 任意でPost Vision Appの設計に必要なOSSや外部ライブラリをインストールし、Post Vision Appに組み込む

ユーザーが任意で使用する、OSSや外部ライブラリのインストール、使用については本SDKでは保証しない。

Post Vision AppコードからWasmファイルを生成

1. **README.md** の手順に従って、Post Vision Appコードのファイル格納場所やファイル名についてMakefileを修正する
2. **README.md** の手順に従って、VS Code UIからターミナルを開き、Wasmビルドするコマンドを実行する
    - Dev Container上にWasmビルド環境用のDockerイメージが作成され、**Makefile** と同階層に.wasm形式のファイルが格納される

ビルド生成物の削除

1. **README.md** の手順に従って、VS Code UIからターミナルを開き、ビルド生成物を削除するコマンドを実行する
    - Wasmビルドによって生成されたファイル(オブジェクトファイル、Wasmファイル)がすべてDev Container上から削除される

ビルド生成物とWasmビルド環境用Dockerイメージの削除

1. **README.md** の手順に従って、VS Code UIからターミナルを開き、ビルド生成物とWasmビルド環境用のDockerイメージを削除するコマンドを実行する
    - Wasmビルドによって生成されたファイル(オブジェクトファイル、Wasmファイル)とWasmビルド環境用のDockerイメージがDev Container上からすべて削除される

Wasmビルドやビルド生成物・ビルド環境用Dockerイメージを削除するコマンドを実行する際に、**README.md** に記載している以外の引数をつけてコマンドを実行すると下記のエラーを返す。

```
ERROR: '<引数>' is unexpected argument.
Please see the document.
```

Post Vision Appインタフェース

Post Vision Appを設計する際は、Post Vision Appのインタフェースとなる関数群の実装が必要になる。サンプルコードには、それらの関数の実装例を載せる。詳細は、別資料のPPLインタフェース仕様を参照。

目標性能

- ユーザビリティ
  - SDKの環境構築完了後、追加のインストール手順なしに「FlatBuffers」のクラス定義ファイルの生成、Wasmビルドができること

制限事項

◆ Wasmビルドについて、C言語、または、C++言語で実装したPost Vision Appコードのみをサポートする

その他特記事項

- SDKに付属する、Post Vision Appの開発に必要なツールのバージョン情報は下記から確認する
  - 「FlatBuffers」: post processディレクトリにある **`README.md`** に記載
  - その他のツール : post processディレクトリにある **`Dockerfile`** に記載

未決定事項

◆ なし