SONY

AITRIOS

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

Version 0.2.0

2023 - 1 - 30

目次

1. 更新履歴

Date	What/Why
2022/11/16	初版作成
2023/01/30	後処理のデバッグ機能の追加。フォルダ構成変更。PDFビルド環境更新。

2. 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
WAMR-IDE	WebAssemblyアプリケーションの実行、デバッグをサポートする統合開発環境
PPL Parameter	PPLライブラリの初期化に使用するパラメータ
LLDB	ソフトウェアデバッガ

3. 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/
  - WebAssembly Micro Runtime(WAMR)
    - https://github.com/bytecodealliance/wasm-micro-runtime/
  - LLDB
    - https://lldb.llvm.org/

4. 想定ユースケース

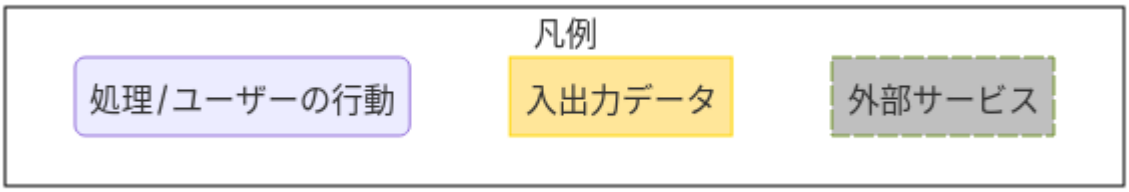
- 後処理を設計、実装したい
- 後処理コードを、エッジAIデバイスにデプロイ可能な形式に変換したい
- 後処理コードを、エッジAIデバイスにデプロイする前にコードのデバッグを行いたい
- エッジAIデバイスにデプロイした後処理コードをデバッグしたい

5. 機能概要、アルゴリズム

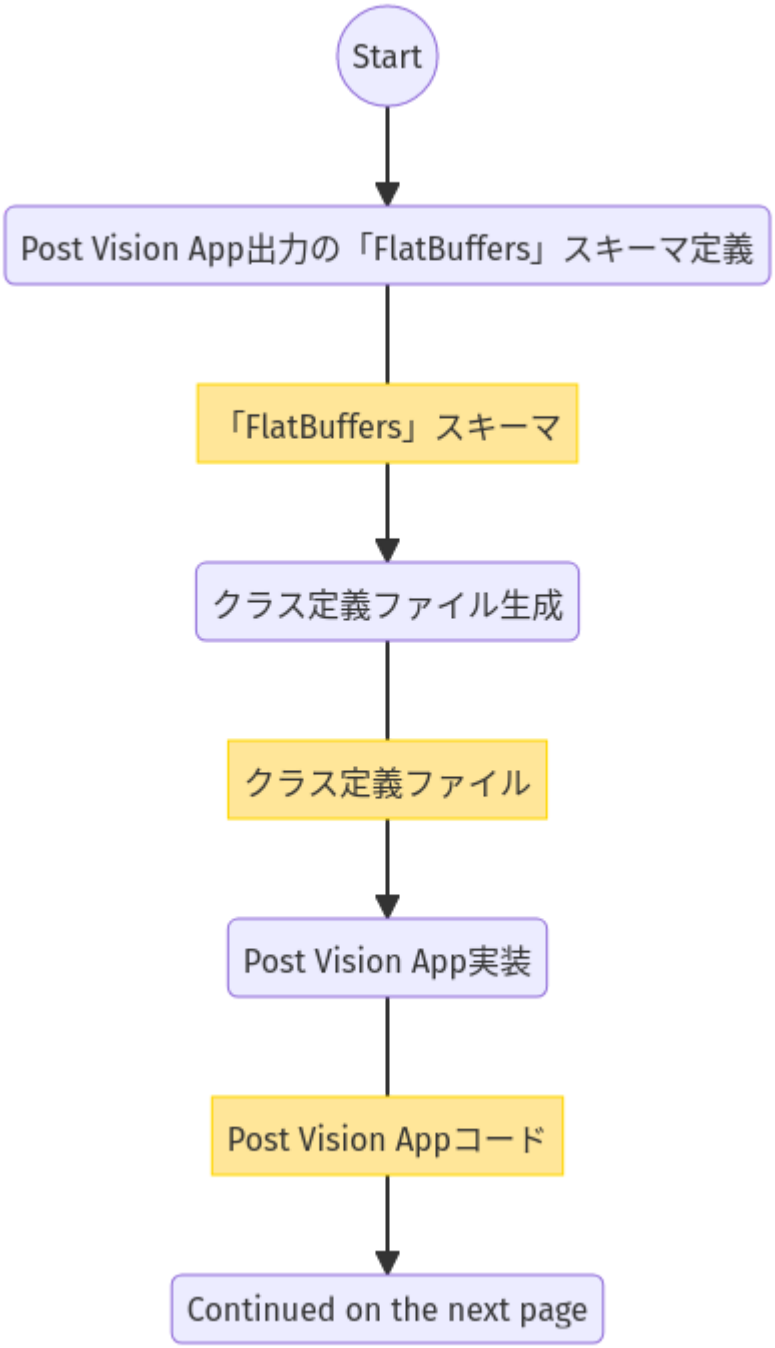
Functional Overview

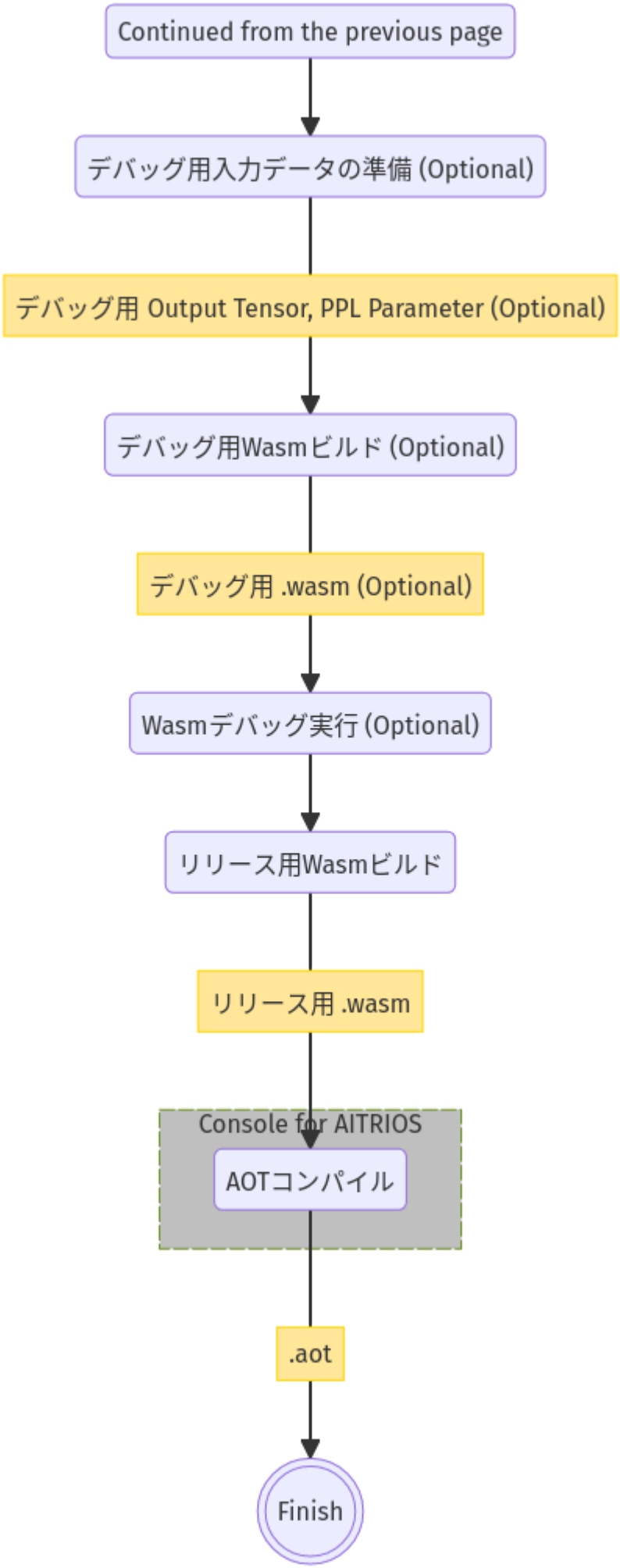
- C言語、または、C++言語でPost Vision Appを設計、実装できる
- Post Vision Appの出力を「FlatBuffers」でシリアライズできる
    - 「FlatBuffers」のスキーマファイルからクラス定義ファイルを生成できる
- C言語、または、C++言語で実装したPost Vision AppをWasmにビルドできる
- Post Vision Appサンプル環境(デバッグ用コードも含む)が提供されている
- Post Vision Appのサンプルコードを使用してWasmビルドができる
- デバッグ用にWasmビルドしたPost Vision Appを、テストアプリを使用しSDK環境上でデバッグ実行できる

Post Vision App作成フロー



フロー




Continued from the previous page
デバッグ用入力データの準備 (Optional)
デバッグ用 Output Tensor, PPL Parameter (Optional)
デバッグ用Wasmビルド (Optional)
デバッグ用 .wasm (Optional)
Wasmデバッグ実行 (Optional)
リリース用Wasmビルド
リリース用 .wasm
Console for AITRIOS
AOTコンパイル
.aot
Finish

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。（デバッグ用ビルドは対象外）

ビルド機能

下記のビルド機能を提供する。

- リリース用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
- デバッグ用Wasmビルド
  エッジAIデバイスにデプロイする前に、コードのデバッグを行うために、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

デバッグ機能

テストアプリを使用したデバッグ実行機能

- LLDBライブラリとWAMR-IDEライブラリとVS Code UIによる下記のWasmデバッグ実行機能を利用できる。
  - breakpointを指定
  - ステップ実行（ステップイン、ステップアウト、ステップオーバー）を指定
  - watch expressionを指定
  - variableを確認
  - call stackを確認
  - ターミナル上でログを確認
- Wasmファイルの処理を呼び出すドライバとして、テストアプリを提供する。
  - Wasmに入力するパラメータ(Output Tensor、PPL Parameter)をテストアプリ実行時に指定できる

WAMR-IDEのproject管理機能は本SDKでは非サポートとする。

デバッグ用サンプルコードで実現できる機能

デバッグ用のサンプルコードを提供する。
デバッグ用のサンプルコードでは、Post Vision Appの出力にデバッグ情報を格納することができる。
仕組みとして本来はPost Vision Appにて処理された推論結果を格納する箇所に、代わりにデバッグ情報を格納することによってデバッグ情報を取得できる。

本機能を利用することで、Post Vision AppをエッジAIデバイスにデプロイした後に推論結果が取得できないなどの推論実行時の予期しない動作に対して、エラー情報を取得することができる。

- Console for AITRIOSで推論実行した場合の例

下記はConsole for AITRIOSにて推論実行した際に取得できる推論結果の例である。

```
{
  "DeviceID": "xxxxx",
  "ModelID": "xxxxx",
  "Image": true,
  "Inferences": [
    {
      "T": "xxxxx",
      "O": "AQAAAA=="
    }
  ],
  "id": "xxxxx",
  "_rid": "xxxxx",
  "_self": "xxxxx",
  "_etag": "xxxxx",
  "_attachments": "xxxxx",
  "_ts": 0
}
```

`"Inferences"` の **`"O"`** に推論結果が格納される代わりに、デバッグ情報が格納される仕組みとなっている。上記では、Base64エンコードされたデバッグ情報 **`"AQAAAA=="`** が格納されている。

6. 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、 `4_prepare_application` ディレクトリの `README.md` にジャンプする
4. `4_prepare_application` 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コードからデバッグ用Wasmファイルを生成

> デバッグ機能を使用する場合のみ、本手順を実施する。

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、デバッグ用Wasmビルドするコマンドを実行する
    - Dev Container上にデバッグ用Wasmビルドとデバッグ環境用のDockerイメージが作成され、`Makefile` と同階層にdebugフォルダが作成され、そのフォルダ内に.wasm形式のファイルが格納される

Wasmファイルのデバッグ用入力パラメータ編集

デバッグ機能を使用する場合のみ、本手順を実施する。

1. `README.md` の手順に従って、テスト実行時の入力パラメータ(Output Tensor、PPL Parameter)を修正する

Wasmファイルをデバッグ実行

デバッグ機能を使用する場合のみ、本手順を実施する。

1. `README.md` の手順に従って、デバッグ実行し、VS Code UIのターミナルでログを確認したり、VS Code UIにてWasmソースコードを開きbreakpointを指定してstack等を確認する

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releaseフォルダが作成され、そのフォルダ内に.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
    - リリース用Wasmビルド、デバッグ用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
    - リリース用Wasmビルド、デバッグ用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.
```

7. 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。

8. 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、Wasmデバッグができること

9. 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

10. 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」: `4_prepare_application` ディレクトリにある `README.md` に記載
  - その他のツール: `4_prepare_application` ディレクトリにある `Dockerfile` に記載

11. 未決定事項

- なし