SONY

Vision and Sensing Application SDK バージョン管理 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
GitHub	ソフトウェアバージョン管理プラットフォーム

参照資料

- ◆ Reference/Related documents（関連資料）
  - ◆ GitHub Document
    - https://docs.github.com/ja

想定ユースケース

- ◆ データセットダウンロード用Jupyter Notebookをバージョン管理したい
- ◆ モデル量子化用Jupyter Notebookをバージョン管理したい
- ◆ 後処理ソースコードをバージョン管理したい

機能概要

Functional Overview

- ◆ 本SDKは、GitHubリポジトリとしてユーザーに提供される
- ◆ ユーザーはGitHubからSDKリポジトリをフォーク(またはクローン)して開発を行う
- ◆ ユーザーはGitHubの機能を利用してバージョン管理を行う

操作性仕様、画面仕様

前提条件

- GitHubアカウントを所有していること

How to start each function

- GitHub上から本SDKのリポジトリにアクセスし、ユーザーの環境に本SDKのリポジトリをフォークまたはクローンする
- チームまたはプロジェクトの運用ルールに従って開発を進める

参考：バージョン管理の運用例

- ◆ 以下、バージョン管理の運用方法の例を記載する
- ◆ チームまたはプロジェクトの運用ルールがある場合はそれに従うこと
- ◆ SDKのフォルダ構成について：
  - ◆ 本SDKは、下記の通り機能種別ごとに独立したフォルダを持つ構成となっている
  - ◆ 各機能は互いに独立しており、個別にバージョン管理することができる

```
/tutorials
    /1_initialize
    /2_prepare_dataset
    /3_prepare_model
    /4_quantize_model
    /5_post_process
    /6_deploy
    /7_evaluate
/.devcontainer
/README.md
```

Table 1. [illegible]

機能種別	管理対象	説明	想定される修正	バージョン管理の目的(例)
prepare dataset	Notebook、設定ファイル	学習用画像データをダウンロードするためのJupyter Notebook	目的に合わせて各種設定値を編集する。(画像のカテゴリや取得枚数など)	画像ダウンロード時に指定したパラメータや設定値を記録しておき、同じ画像データで学習を再現したい場合などに参照できるようにしておく。

機能種別	管理対象	説明	想定される修正	バージョン管理の目的(例)
quantize model	Notebook、設定ファイル	自前のAIモデルを量子化するためのJupyter Notebook	目的に合わせて各種設定値を編集する。	モデルを量子化した際のパラメータなどの情報を記録しておき、 同じパラメータで学習を再現したい場合などに参照できるようにしておく。
post process	post processソース一式、Makefile	モデルの後処理を記述するソースコード、 および、Wasm形式にコンパイルするためのビルドファイル	後処理のロジックを開発する。	更新履歴を残しておくことで開発の効率を向上させる。 また、チームでの開発を容易にする。

- ◆ ブランチ
  - ◆ ブランチを作成することで、複数の機能開発を同時に進めることができる
  - ◆ また、prepare dataset、quantize model、post processの編集をそれぞれ別ブランチで行うことにより、
    独立してバージョン管理することもできる
    - ブランチ運用例：

```
main
     |-- feature/post_process/object_detection_xxx (1)
     |-- feature/post_process/image_classification_xxx (2)
     |-- feature/quantize/xxx (3)
     |-- feature/prepare_dataset/xxx (4)
     |-- bugfix/XXX (5)
```

(1) Object Detectionモデル向け機能開発ブランチ

(2) Image Classificationモデル向け機能開発ブランチ

(3) 量子化管理用ブランチ

(4) データセットダウンロード管理用ブランチ

(5) バグ修正用ブランチ

◆ コミット履歴

◆ ファイルの変更内容を、保存したいタイミングでコミットすることで、変更履歴として後から参照できる

- 運用例：
  - Notebook実行時の情報を保存する
    1. **`configuration.json`** の設定値やNotebookを編集した状態でGitブランチにコミットする
       - これにより、Notebookを過去と同じパラメータ設定で再実行したい場合に、
         コミット履歴から設定を参照することができる
    2. 加えて、入力に使用したデータの情報をコミットに紐づけておくことで、 同一条件 (入力データ、パラメータ) でのNotebook実行を再現することができる
       - 入力データとコミットを紐づける方法はユーザーに委ねられるが、簡単な例としてはコミットにコミットメッセージとして記載しておく方法がある
         - コミットメッセージ例 (quantize modelの場合) :

```
Quantization test
    description: xxxxxx
    input model: <url_to_model_resistry>
    dataset: <url_to_dataset_resistry>
    ....

# Please enter the commit message for your
changes. Lines starting
# with '#' will be ignored, and an empty
message aborts the commit.
#
# Committer: XXXXXX
#
# On branch feature/quantize/xxx
# Changes to be committed:
#       modified:   configuration.json
```

◆ タグ

- コミットに対してタグを付与しておくことで、必要なバージョンへのアクセスが容易になる
  - Gitコマンドによるタグ付け：

```
git tag -a [tag name] -m 'tag comment' [commit id]
```

目標性能

◆ なし

制限事項

◆ なし

その他特記事項

◆ なし

未決定事項

◆ なし