

AI 実装支援プラットフォームの開発

根尾敦 松下和寛 田浦健 小林伊織 大宮英典

株式会社日立製作所

1. はじめに

人物を対象として，AI(Artificial Intelligence)を用いる画像認識システムの開発には，個人情報保護法に準じた画像の取り扱いに加え[1]，アクセラレータを用いた処理の高速化など専門知識を要するシステム開発が課題となる．画像の取扱いは，取扱うネットワークや，取扱者を限定する組織も多いが[2]，取扱者が専門知識を備えているとは限らず，システム導入に課題を抱える現場も多い．こうした課題に対して，本研究では，現場サイドでの画像認識システム開発の支援が有効と考え，専門知識を不要とするノンプログラミングツールを開発した．以下，開発したツールを「HiLEC」と表記する．HiLECはNode-REDとOpenVINOを含み，マウス操作により，GPUアクセラレートにも対応した現場サイドでの画像認識システム構築を実現する．

2. 開発方針

HiLEC開発において，全てを独自に開発する場合には以下の点が懸念される．

- (1) 開発工数がかかる．
- (2) 業界標準から離れ，他ツールとの連携が困難な孤立した環境を構築する可能性がある．

こうした中，ノンプログラミングツールは，簡易的な日程管理用途からシステム開発用途まで，有償に限らず無償のOSSも幅広く存在する．引用数の高いOSSの活用は，工数削減と共に，標準性の確保が可能と考える．そこで，開発工数を抑制し，業界標準になり得るノンプログラミングツールを目指すため，開発者数とコミット数が多く，Raspbianでも推奨ソフトウェアとされるほど認知度の高いNode-REDの活用を考える．Node-RED活用にあたっては，活用可能な部分と，独自開発が必要な部分に分けて検討する必要があるため，AI実装支援ツール「HiLEC」に対するNode-REDの課題抽出から始める．

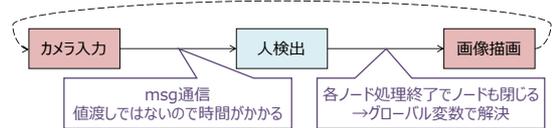
3. 開発課題

Node-REDを活用したHiLECの開発には3つの課題がある．

Node-REDは，システムを設計するUI(User Interface)のフロントエンドと，UIから命令を受けて処理を実行するバックエンドから構成される．Node-REDのバックエンドは「Node.js」により動作する．しかしHiLECはAI実装を目的としており，多くのAIフレームワークが開発言語としているPythonでの動作が必要になる．このため，Node-REDのバックエンドは使用できない．これが1つ目の課題となる．

2つ目の課題を図1を用いて説明する．機能をノードとして保有し，各ノードを結線することでシステムを構築するNode-REDでは，ノード間のデータの受け渡しをメッセージ通信により実現する．一般的なプログラミングのデータ渡しや参照と異なり，ノード間ではメッセージ通信が実施されるため，ノード間に通信コストが発生し，処理速度低下やリソース消費増大が懸念される．また，各ノードは個別の処理が終了する度に閉じられるため，ストリーミング処理のように繰り返して処理を行う場合にはノード毎に処理した変数を保持できない．これにより，グローバル変数の定義や，ノード間の変数のやりとりが一般的なプログラミングに比較して増加してしまう．またノード内の処理においても，AI処理のため「Node.js」とPython領域間にデータ通信が必要となりノード内においても通信コストが発生する．

●ノード間通信と処理



●ノード内処理

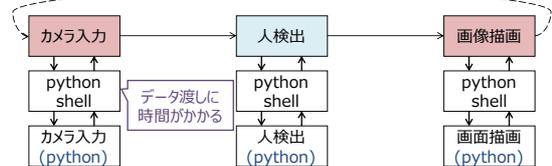


図1 Node内外でのデータ通信の課題

Development of AI implementation support platform
 Atsushi Neo, Kazuhiro Matsushita, Ken Taura, Iori Kobayashi, Hidenori, Omiya, Hitachi, Ltd.
 185-8601, Tokyo, Japan

3つ目の課題を説明する。Node-REDはユーザビリティを重視したツールであるため、設計後にフロントエンドから容易にシステムを実行できる。しかし設計基盤と実行基盤が共通になっていることで、設計後にも設計基盤を含むNode-REDを立ち上げてシステムを動作させる必要があり、組込みシステムへの適用を考えた際には効率が悪い。

4. 開発結果

0章で挙げた3つの課題を解決するために、図2に示す構成を考案した。まずユーザビリティを追求するため、設計を行うフロントエンドはNode-REDを引き続き採用する。一方で処理を行うバックエンドは、Pythonでまとめて実行する。この構成での動作を実現するためNode-RED上のノードは機能の実体を持たずに、どのような機能であるかを示す機能情報のみを持つ。各ノードは後段のノードに、機能情報を出力する。詳細動作を具体例を挙げて説明する。初段の「カメラ入力」は、後段の「人検出」に「カメラ入力」という機能情報を出力する。「人検出」は前段からの入力に加えて、当該ノードの「人検出」という機能情報を後段の「画像描画」に出力する。システムの最後には「実行ノード」を配置する。このため「実行ノード」にはシステムの経路を含む全機能の情報が出力される。この情報を構成情報として、実行ノードはバックエンドに出力し、システムコールにより所定のpythonプログラムを実行する。構成情報は、機能の配置と機能を示す情報のみを保有する。つまり機能の実体はない。一般的なNode-REDであればノード間で通信される情報は処理の戻り値となるが、本ツールにおいては機能情報のみとなり、機能の実体は全てバックエンド側に構成情報が渡された後に実行される。

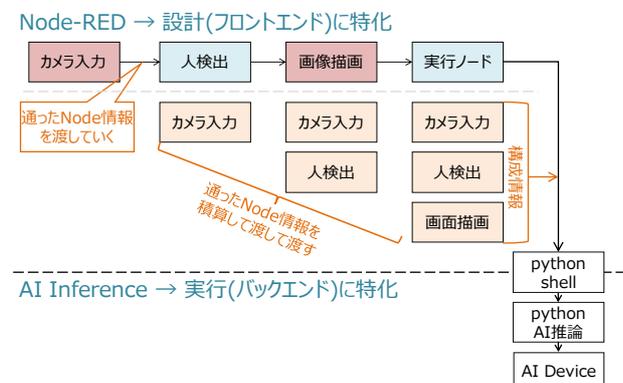


図2 HiLEC開発のポイント

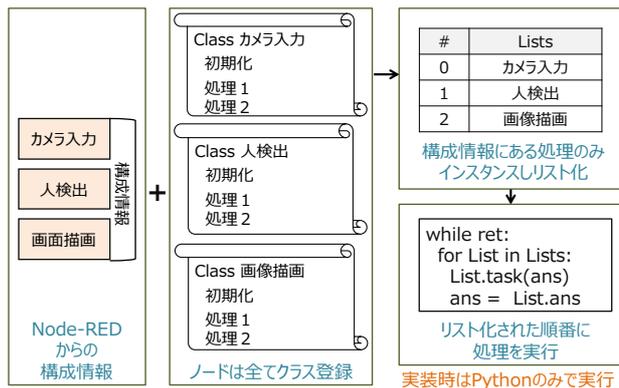


図3 HiLECバックエンド

機能の実体を実行するバックエンドの処理を図3を用いて説明する。バックエンドには構成情報記載の機能を含め、フロントエンドのNode-REDに登録されるノード全ての機能をpythonのオブジェクトとして予め登録する。実行時には構成情報記載の機能を順にインスタンスし、リスト形式で保有する。そしてリスト化された順に処理を繰り返す。

5. 評価

Node-REDに対して、HiLECはバックエンドをPython動作とし、フロントエンドと、バックエンドを構成情報により関連付けた。これにより、フロントエンドはNode-REDのユーザビリティを維持しつつ、バックエンドは構成情報を読み込むpythonのみの処理となったことで、GPUなどのアクセラレータを用いた高速化処理も容易に設計できる。設計後の設計情報は全て構成情報に保有されるため、設計後にはフロントエンドに採用するNode-REDの立ち上げは不要となる。

6. まとめ

AI画像認識システムを設計可能とするノンプログラミングツール「HiLEC」を開発した。これにより、AIシステム設計において、専門知識を不要とした上で設計性・実装性向上を実現した。

参考文献

- [1] “個人情報の保護に関する基本方針”, 個人情報保護委員会
- [2] 中野潔, 浅野幸治, “防犯カメラについての公的なガイドラインの比較における一考察”, 情報処理学会研究報告マルチメディア通信と分散処理 (DPS), pp. 37-42, 2005