開発レポート　Development Report

カメラ認識技術検証アプリ「Vision Studio」の開発

― 色検知・追跡・ライン・アーム接近・コード読取の統合 GUI ―

Vision Studio: An Integrated GUI for Camera-Based Color Detection, Tracking, Line Following, Arm Approach and Code Reading

レポート番号	MTIC-DR-2026-004
団体	NPO 法人松戸技術革新センター M-TIC　松戸ロボット研究会
著者	大城 公平
発行日	2026 年 10 月 25 日
対象	Vision Studio v0.2.3（ソフトウェア）
分類	開発レポート／画像認識ソフト
機体状態の基準日	2026 年 9 月 21 日

本レポートの数値と機体状態について　本検証系は研究開発段階にあり随時変更されるため、記載の仕様は基準日時点のものであり、現在の実機と一致することを保証しない。数値には種別を付す：**公称**＝カタログ・設計値、**実測**＝実際に計測した値（記載桁が測定分解能）、**概算**＝目測・推定値。

概要

ロボットにカメラ認識を導入するには、認識の状態を人が確認しながら調整し、その結果をログや制御系へ渡せる道具が必要である。「Vision Studio」（正式名称：松戸技術革新センター カメラ認識技術検証アプリケーション）は、色検知・物体追跡・ライントレース・アーム接近・QR／バーコード読取の 5 モードを 1 つの GUI で切り替え、認識結果を JSONL ログ・UDP・シリアルで出力するアプリである。プログラムが得意でない人でも GUI で操作できることと、Raspberry Pi 4B で動く軽さを優先し、pillow 等の追加ライブラリに依存しない構成とした。アーム接近モードは、水平多関節（SCARA 型）アームの先端付近のカメラで対象物を捉え、画面内の位置と大きさからアームを対象へ近づけるための情報を出力する。ライントレース専用のソフトは別レポート（MTIC-DR-2026-006）で報告する。各モードの認識精度と処理速度はまだ計測しておらず、次段階で搭載先の CWM・CBV 上で計測してデータをもとに評価したい。

1. 背景と目的

1.1 開発背景

M-TIC が研究開発しているピッキングロボット CWM と運搬ロボット CBV は、Raspberry Pi 4B と Pico 3 台（CAN 接続）で構成され、カメラを搭載している。物体検知・ライントレース・物体追跡・アームの物体接近をカメラで行えるかを検証するにあたり、認識状態を目で見て調整でき、結果を制御系に渡せる GUI ツールが必要になった。

1.2 目的と目標

ロボット本体を動かすことよりも、カメラ認識の状態をログや制御系に反映できる出力まで落とし込むことを目的とする。GUI で操作でき、Raspberry Pi 4B で動く軽さを条件とする。

目標：5 つの認識モードを 1 つの GUI で切り替えられ、認識結果が JSONL・UDP・シリアルのいずれでも出力されること。

2. システム構成

2.1 開発環境と基本的な作り方

開発言語は Python、開発環境は Thonny である（PC で書いたものをそのまま Raspberry Pi 4B で動かせる点を重視した）。画像処理は OpenCV、画面は Python 標準の Tkinter で組み、認識結果は UDP・シリアル・JSONL ログの 3 つの経路で外に出せるようにした。機体制御ソフトは別プログラムのままにしてあるので、認識側だけを差し替えられる（図 1・図 2）。



図 1　開発環境と基本構成



図 2　Vision Studio の処理の流れ

※ 本レポートでは処理の考え方と構成のみを示し、ソースコードおよび調整値は掲載しない。

2.2 Vision Studio 本体

項目	内容
正式名称	松戸技術革新センター カメラ認識技術検証アプリケーション（略称 Vision Studio）
版	v0.2.3（2026-09-21）
実行環境	Python（Thonny）。PC で開発、Raspberry Pi 4B で実行。pillow などの追加ライブラリを使わない軽量構成
モード	① 色検知 ②物体追跡 ③ライントレース ④アーム接近 ⑤ QR／バーコード読取
出力	JSONL ログ／UDP／シリアル
構成方針	機体制御ソフトとは別プロセス。UDP で状態を渡し、機体制御ソフトは改変しない
搭載先	CWM・CBV（ともに Raspberry Pi 4B、最新 OS）。CTR は対象外

2.3 アーム接近モードの考え方

ピッキングロボット CWM の水平多関節（SCARA 型）アームで物をつかむには、対象物がアームに対して「左右どちらに・どれくらい離れて」あるかが必要になる。アーム接近モードでは、アーム先端付近に取り付けたカメラの画像から対象物を色や形で検出し、画面の中心に対する対象物の位置のずれと、映っている大きさを求める。位置のずれはアームを旋回させる方向の目安に、大きさは対象までの近さの目安に

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

なる（近いほど大きく映る）。これらを UDP またはシリアルで出力し、アームの制御側で使う（図 3・図 4）。



図 3　アーム接近モードの処理の流れ

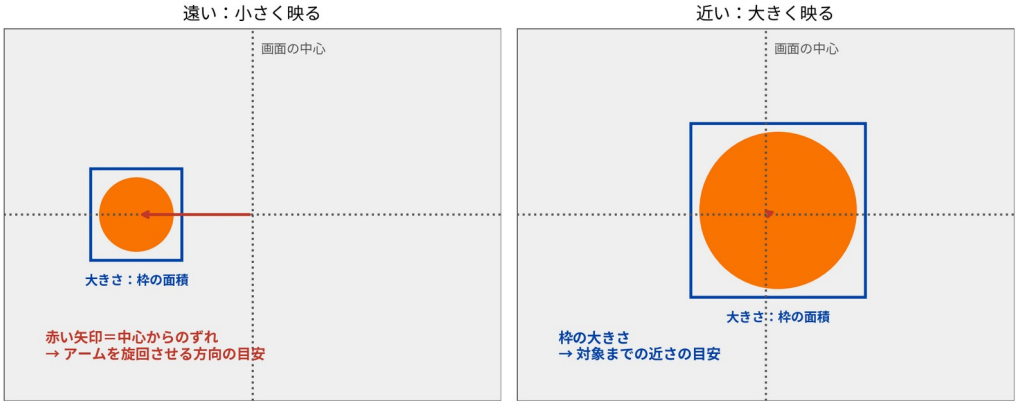


図 4　画面中心からのずれと枠の大きさ（概念図）。ずれは旋回方向の、大きさは近さの目安になる

2.4 認識対象と手法の方針

- 認識したいもの：自前で撮影・学習させる物体 5 種類程度、ライン、走行経路上の人、障害物、QR コード、バーコード
- 物体学習は Teachable Machine → TFLite の流れで進める（YOLOv5 はラズパイ 4B では重いと判断）。学習型の物体検知は M-TIC Detect（MTIC-DR-2026-005）へ分離
- 認識結果を反映する先は、ラズパイの GUI から Pico 3 台へ CAN で指令する構成（CAN は MTIC-TR-2026-001 参照）

3. 検証方法と結果

3.1 検証方法

PC の Thonny 上で USB カメラを接続し、5 モードの切替と各モードの認識表示、3 種の出力を確認した。

3.2 結果

項目	結果
5 モードの切替・表示	PC 上で動作確認（v0.2.3）
JSONL／UDP／シリアル出力	PC 上で出力を確認
アーム接近の位置ずれ・大きさの出力	PC 上で出力を確認。アームを実際に動かした検証は未実施

各モードの認識精度と、Raspberry Pi 4B での処理速度は未計測である（5.1 節）。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

4. 課題と考察

- **実機での確認：**PC 上の確認にとどまる。ラズパイ 4B での処理速度、照明条件による色検知の変動が主要な不確定要素。
- **学習型認識の分離：**Vision Studio 単体では自作学習を持たず、学習型は M-TIC Detect に分けた。両アプリの出力を機体側でどう統合するかは未設計。
- **カメラ切替の重さ：**カメラ切替が重い問題に対し、共通カメラモジュールを他アプリで先行導入済みであり、本アプリへの適用は次版で行う。

5. 今後の展望

5.1 検証計画

計測項目	方法	判定基準
処理速度	ラズパイ 4B で各モードの fps を記録	制御に必要な fps を満たす
色検知の安定性	照明 3 条件（昼・夜・蛍光灯）で検出率を記録	各条件で検出率を記録し基準を設定
アーム接近	SCARA 型アームに搭載し、対象物への接近・把持を 10 回試行	成功回数／10、失敗の原因を記録
QR／バーコード	距離・角度を変えて読取可否を記録	読取可能距離・角度

5.2 応用の可能性

将来的には販売できる形を目指す。カメラ認識を GUI で確認しながら調整できる構成は、他の機体や用途での認識技術の検証にも使える可能性がある。

5.3 カメラ認識技術の統合（構想）

M-TIC で開発してきたカメラ認識ソフト（床面認識・ライン認識・物体検知・色や形の認識）を組み合わせて 1 つの自動運転に使う構想は、CWM_CamLine のレポート（MTIC-DR-2026-006）の 5.3 節にまとめている。Vision Studio は、その中で色・形・QR など「何を見るか」を切り替える道具として使う。

6. まとめ

5 つの認識モードと 3 種の出力を 1 つの GUI に統合した Vision Studio v0.2.3 を開発し、PC 上で動作を確認した。アーム接近モードでは、対象物の位置のずれと大きさをアーム制御の目安として出力できる。認識精度・処理速度・アームへの搭載はいずれもまだ確かめておらず、次段階で CWM・CBV 上で計測したい。

参考資料

OpenCV ドキュメント（opencv.org）

Google, “Teachable Machine” ／ TensorFlow Lite ドキュメント

改訂履歴　初版（2026-09-30）　2026-10-25　文章を見直し、誤記を訂正。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.