

2025年度

学士論文

論題

スマートフォン主導の開放語彙物体検出に基づく
自作モデル運用基盤の設計と実装

指導教員 孟林教授

立命館大学 理工学部 電子情報工学科

学籍番号 2290220041-3
氏名 後藤 晴貴

論文要旨

AIが広く普及した現代において、対話型AIは一般ユーザに浸透している一方、画像認識は依然として専門知識を要し、エンジニア主体の領域に留まっている。本研究は一般ユーザを主対象とし、データ収集からアノテーション、学習、評価、利用までを一貫して支援し、個々の利用状況に特化した画像認識モデルの調整・ファインチューニングを可能にするアプリケーションを開発した。実装はプロトタイプとして公開し、フロントエンドにReact Native + Expoを採用してAndroid/iOS/Webで統一的に動作するUIを提供、直感的なドラッグ&ドロップのラベリング、リアルタイム推論、学習進捗の可視化、モデル管理などの機能を統合した。バックエンドはFastAPIを用い、Ultralytics YOLOを中心とする検出・学習エンジン、学習履歴・メトリクス取得、データセット分析APIを実装し、非同期学習やモデルの保存・読み込みを含むワークフローを提供する。開発運用面ではMakefileによるセットアップ／起動／テストの自動化、pnpmを用いたクロスプラットフォーム開発、OpenAPIによるエンドポイント記述を整備し、再現性と保守性を高めた。本システムにより、プログラミング経験が乏しいユーザでも少量の自前データから用途特化モデルを反復的に改善でき、画像認識活用の敷居を下げる。ケースとして料理画像検出を対象に、UI内のデータ収集・ラベリングから学習、精度の可視化までを一連の操作で完結できることを示し、一般ユーザによるモデルカスタマイズの実用可能性を示唆する。

目次

1	はじめに	1
2	関連研究	3
2.1	物体検出モデル	3
2.1.1	YOLO-World	3
3	提案手法	5
3.1	設計方針と構成	5
3.2	YOLO-World の運用	5
3.3	データ収集・ラベリング・学習	5
3.4	UI フロー（スマホ中心）と操作例	5
3.5	スマホ最適化と制約	10
4	語彙登録のみでの検出成立性の検証	11
4.1	目的	11
4.2	操作フローと条件	11
4.3	評価項目	11
4.4	ケーススタディ（smartphone）	11
4.5	考察と制約	12
5	手動ラベリングとファインチューニングによる検出成立性の検証	13
5.1	目的	13
5.2	前提と環境	13
5.3	手順	13
5.4	評価設計	13
5.5	実装上の要点	14
5.6	実験結果	14
5.7	期待と限界	14
5.8	今後の改善	14
6	まとめ	15
A	実験結果	19
A.1	領域検出精度の実験結果一覧	19
A.2	測定の実験結果一覧	19

図目次

3.1 提案 UI の反復ループ (1/4) : 撮影→初回検出→語彙確認/追加	6
3.2 提案 UI の反復ループ (2/4) : 再検出→ラベリング→学習起動	7
3.3 提案 UI の反復ループ (3/4) : 語彙・履歴・モデル管理	8
3.4 提案 UI の反復ループ (4/4) : データ・性能の確認と再反復	9
A.1 球冠ベースの結果画像 (ユニオンモデル)	20
A.2 関数ベースの結果画像 (ユニオンモデル)	20
A.3 球冠ベースの結果画像 (ユニオンモデル 2)	21
A.4 関数ベースの結果画像 (ユニオンモデル 2)	21

表目次

A.1 学習結果(ユニオンモデル)	20
A.2 学習結果(ユニオンモデル2)	21
A.3 測定結果(ユニオンモデル)	22
A.4 測定結果(ユニオンモデル2)	23

1

はじめに

対話型 AI の普及により、一般ユーザが自然言語によって高度な情報処理を日常的に活用する時代が到来した。一方で、画像認識をはじめとするコンピュータビジョン (CV) は、データ収集、アノテーション、学習・評価、運用を含む一連の工程が分断されやすく、専門的な知識・ツール群を横断的に扱う必要があることから、依然として一般ユーザにとって参入障壁が高い。特に、用途特化のモデルを自分で調整（チューニング／ファインチューニング）し、反復的に改善していくためには、学習用データの拡充、失敗の可視化、改善仮説の検証といった実務的ワークフローが不可欠である。

本研究では、こうしたギャップを解消し、一般ユーザが「自分の目的に合った画像認識モデル」を自力で構築・調整できる環境を提供することを目的として、エンドツーエンドのモデル管理アプリケーション「Dish Detection」を開発した。本システムは、スマートフォン／PC のいずれからでも利用可能なクロスプラットフォーム UI (React Native + Expo) と、高性能な Web API (FastAPI) を組み合わせ、データ収集・ラベリング・学習・推論・評価・モデル運用までを一貫して支援する。具体的には、カメラやギャラリーからの画像取得、ドラッグ＆ドロップによる直感的なアノテーション、Ultralytics YOLO を用いたリアルタイム物体検出、学習の非同期実行と進捗監視、履歴の可視化、データセット分析、モデルの保存・切替といった機能を統合し、一般ユーザでも試行錯誤を通じてモデルを改善できる実用的なワークフローを実現した。

運用面では、限られた計算資源でも現実的に扱えるよう、学習をバックグラウンドで非同期実行し、UI をブロックしない設計とした。さらに、再現性と保守性を高めるため、Makefile によるセットアップ・起動・テストの自動化、パッケージマネージャ (pnpm) による依存関係管理、OpenAPI によるエンドポイント定義の明示化を行っている。これにより、ユーザは最小限の初期設定で環境を整え、反復的なモデル改善に集中できる。

適用領域としては、料理画像を例題に据え、器や料理種別に応じた検出のしやすさ、データの集め方、モデルの差し替えやクラス管理など、実務的な観点からの検討を行った。用途特化の小規模データから出発し、UI 上でのアノテーションと学習、可視化により改善ポイントを特定しながら、ユーザ自身の目的に合わせたモデルを段階的に洗練させることが可能である。これにより、画像認識活用の敷居を下げ、一般ユーザ主導の”現場適合”モデルの創出を後押しする。

本稿の主な貢献を以下に示す。

- データ収集から学習・評価・運用までを統合した一般ユーザ向け CV モデル管理アプリケーションの設計・実装
- クロスプラットフォーム UI 上での直感的ラベリングとリアルタイム物体検出の統合による反復改善の促進
- 学習の非同期実行、進捗・履歴・メトリクスの可視化、モデル管理機能の一体化による実用的ワークフローの提供

- Makefile, pnpm, OpenAPI 等を用いた再現性の高い開発運用体制の整備

本論文では 2 章で背景および関連研究について述べ、3 章で提案システムの設計方針と機能構成を示す。4 章では語彙登録のみでの検出成立性をスマートフォンから検証し、5 章では手動ラベリングとファインチューニングによる検出成立性の検証計画（結果は未掲載）を示す。6 章ではまとめと今後の課題（軽量化・最適化、拡張可能性、運用上の安全性・信頼性など）について議論する。

2

関連研究

2.1 物体検出モデル

従来の物体検出は COCO などの固定語彙 (close-set) を前提としており、学習時に定義したカテゴリのみに限定されるという制約がある。一方、実環境では「未学習カテゴリ」を含む開放語彙 (open-vocabulary) への拡張が重要である。YOLO 系列は Backbone・Neck・Head からなる一段 (one-stage) 検出器として高い効率を示してきたが、語彙の固定という制約が実用展開のボトルネックとなってきた。

2.1.1 YOLO-World

YOLO-World は、従来 YOLO の効率性を維持しつつ、視覚 - 言語モデリングによって開放語彙検出を実現した検出器である [1]。その中核は、(1) テキスト埋め込みと画像特徴を結合するための再パラメータ化可能な Vision-Language PAN (RepVL-PAN)、(2) 検出データ・グラウンディング・画像テキストの各データを統一的に扱う領域 - テキスト対 (region - text) に基づく大規模事前学習、(3) 推論時の効率を高める「prompt-then-detect (事前語彙化) パラダイム」にある。

まず、学習時は CLIP 系テキストエンコーダで得たテキスト埋め込みを RepVL-PAN に導入し、画像特徴と語彙表現を相互作用させる。推論時にはテキストエンコーダを除去し、オフラインで事前計算したテキスト埋め込みを Neck に再パラメータ化して埋め込むため、実行時コストを抑えつつ開放語彙に対応できる。RepVL-PAN の T-CSPLayer 再パラメータ化の一例は次式で与えられ、 1×1 畳み込みの重みとしてテキスト埋め込みを吸収することで、言語条件付けを含む計算を単純化する（付録記述に基づく）：

$$X' = X \odot \text{Sigmoid}(\max(\text{Conv}(X, W), \dim = 1)), \quad (2.1)$$

ここで X は画像特徴、 W はテキスト埋め込み由來の畳み込み重み、 \odot は要素ごとの積を表す。

学習スキームとしては、領域 - テキスト対に基づくコントラスト学習を大規模データで行う。実データ (Objects365 等) に加え、CC3M などの画像テキストデータから、名詞抽出 → 擬似ボックス生成 (GLIP 等) → CLIP による再スコアリングと NMS/閾値フィルタリングという自動ラベリングパイプラインで領域 - テキスト対を構築し、開放語彙能力を強化する。小型モデル (YOLO-World-S) に対しては、高品質アノテーションや適量の擬似ラベルを組み合わせることでゼロショット性能が向上することが示されている。

性能面では、LVISにおいて 35.4 AP かつ V100 上で 52 FPS を達成し (TensorRT なし)，同規模の既存手法に対して精度・速度のバランスで優位性を示す。また、学習後は「事前語彙化 (offline vocabulary)」によりカテゴリ埋め込みをモデル重みに取り込み、エッジ展開時のテキストエンコーダ依存を排除する。さらに、COCO のような固定語彙タスクへ移行する際は、RepVL-PAN の言語関連層を取り除き、従来 YOLO と同等の運用効率で微調整

可能である。総じて、YOLO-World は固定語彙検出と開放語彙検出の橋渡しを行い、汎用実応用（ゼロショット検出、参照物体検出、オープン語彙インスタンスセグメンテーション等）に適した現実的なデプロイ手段を与える。

3

提案手法

本研究の目的は、スマートフォンを含む汎用端末のみでデータ収集からラベリング、学習、評価、運用までを一気通貫に反復できる実用システムを構築し、非専門家でも短時間で自作の用途特化モデルを育てられることを示す点にある。中核には開放語彙検出器である YOLO-World[1] を据え、事前語彙化（prompt-then-detect）によって実行時の言語エンコーダ依存を排除し、軽量・高速な推論を維持する。

3.1 設計方針と構成

フロントエンドは React Native + Expo により Android/iOS/Web を單一コードベースで提供し、タブ（Detection / Labeling / Training / Models / Analytics）に機能を整理する。バックエンドは FastAPI で統一し、検出・語彙管理・学習・履歴・可視化・データ分析の API を備える。ユーザは語彙を自ら定義・追加し、必要データを小刻みに収集・注釈付けして学習をトリガし、結果を見ながら再収集・再学習を繰り返す。

3.2 YOLO-World の運用

YOLO-World は視覚 - 言語モデリングにより、ユーザ定義語彙での開放語彙検出を実現する。POST /model/classes で登録した語彙は custom_vocab.json に永続化され、モデルのクラス埋め込みへ反映される。推論は/detect で実行し、バウンディングボックス・クラス・スコアと描画済み画像を返す。事前語彙化により、推論時はオフラインで固定した語彙埋め込みを用い、モバイルでも実用的なレイテンシを確保する。

3.3 データ収集・ラベリング・学習

Labeling タブで作成したアノテーションは YOLO 形式で保存し（training_data/ 配下）、/training/start または /training/start-async で微調整を起動する。既定は CPU 実行だが、CUDA 対応マシンでは設定により GPU（device='cuda'）で学習・推論が可能である。完了時には best.pt を自動ロードし、/training/status で進捗を可視化する。/models/*群でモデル一覧・切替・バックアップ・検証が可能で、/training/history と /training/metrics/{run_name} から学習履歴・時系列メトリクスを取得し UI で Plotly 描画する。

3.4 UI フロー（スマホ中心）と操作例

本節では、スマートフォン想定の最短反復ループ（撮影→検出→語彙追加→再検出→ラベリング→学習→評価）を画面遷移で示す。各ページに 4 枚ずつ配置する。

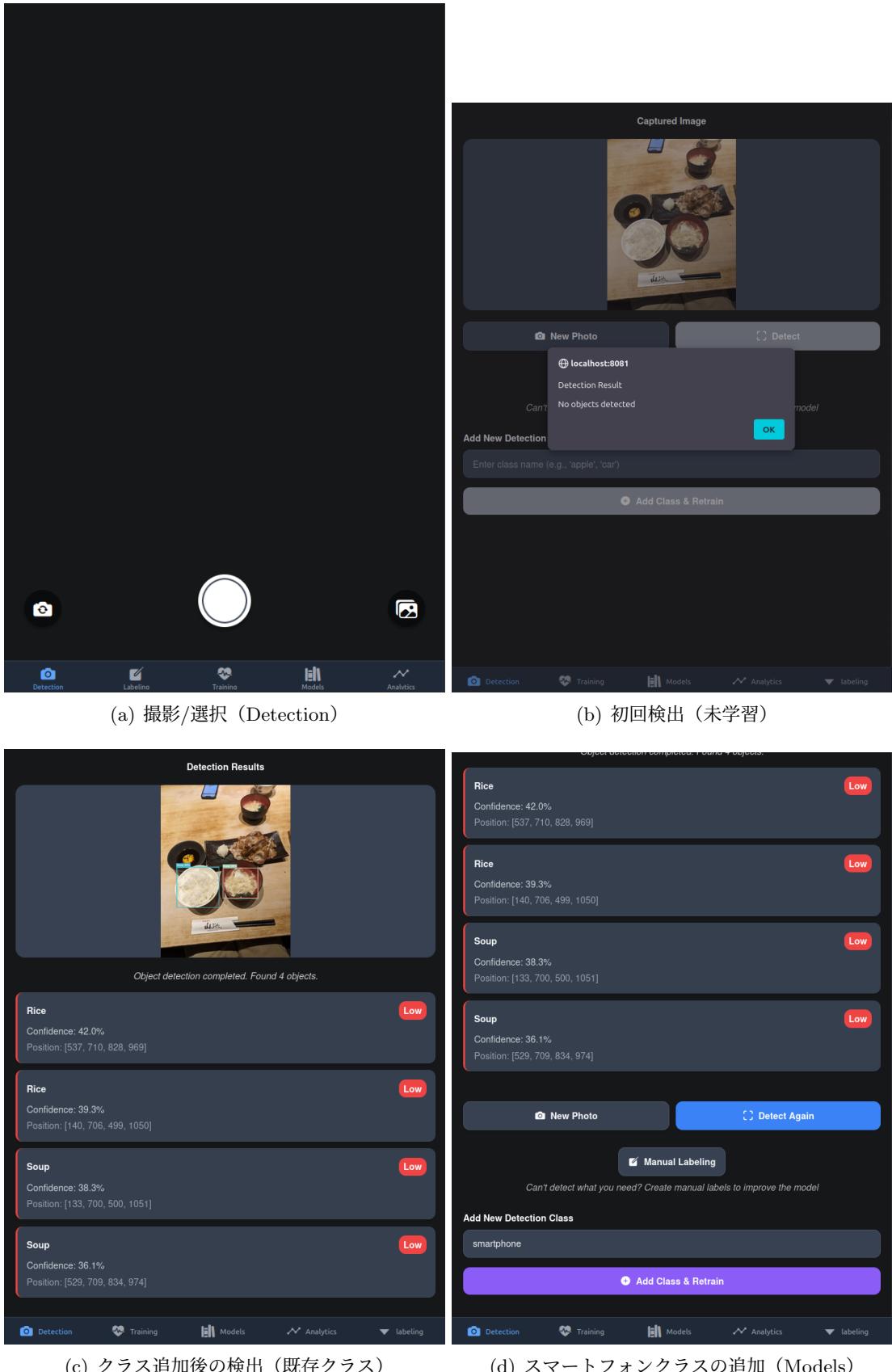
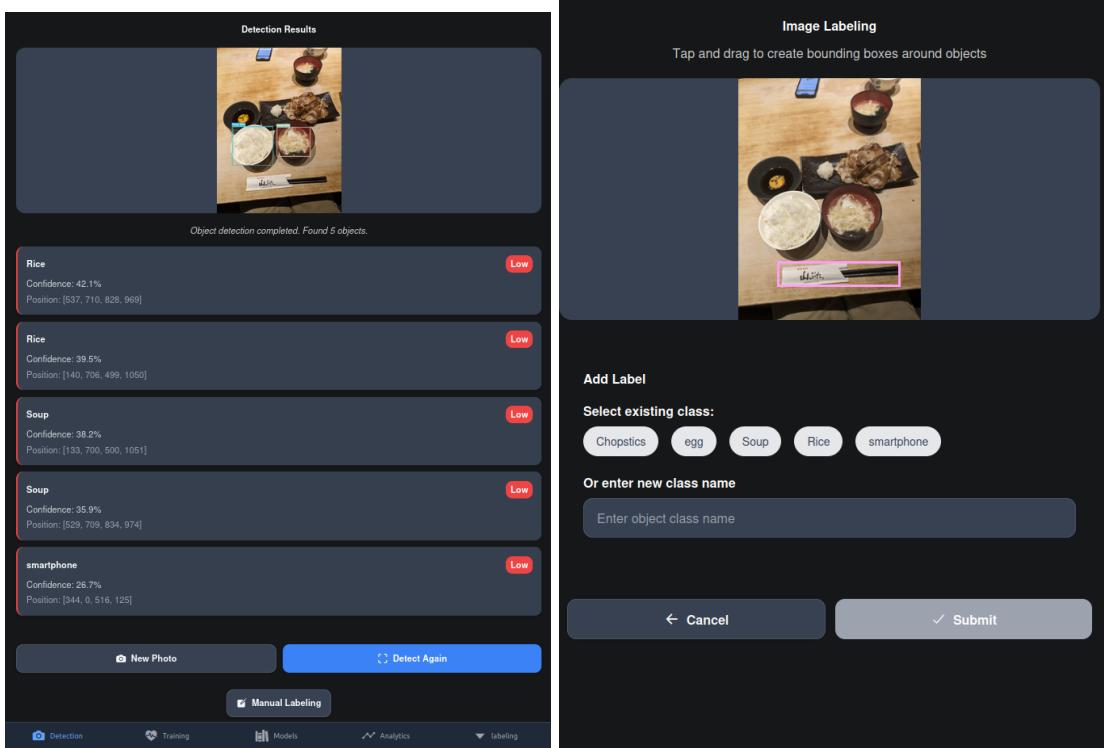


図 3.1: 提案 UI の反復ループ (1/4) :撮影→初回検出→語彙確認/追加



(a) smartphone クラスで検出成功

(b) マニュアルラベリング

Training Center
Comprehensive model training management

Model Training

Training Data Statistics

Images	Labels	Classes
0	0	0

No training data

Training Configuration

Number of Epochs
50

- Higher epochs = better accuracy but longer training time
- Recommended: 50-100 epochs for most cases
- Training time: ~1-5 minutes per 10 epochs

Start Fine-tuning

How to Improve Your Model

- Capture images of objects that aren't detected well
- Use "Manual Labeling" to create bounding boxes and labels

(c) ファインチューニング開始

(d) 学習の起動 (Training)

図 3.2: 提案 UI の反復ループ (2/4) : 再検出→ラベリング→学習起動

Training Center
Comprehensive model training management

Model Management
Manage detection classes for object recognition

Model Information

- Model Path: ./yolov8s-world.pt
- Vocabulary File: custom_vocab.json
- Total Classes: 11

Add New Class

Enter class name (e.g., 'apple', 'car', 'person')

Detection Classes (11)

		Refresh
person	#1	Remove
Road	#2	Remove
car	#3	Remove

Analytics

Training Analytics

Current Status

- Ready

Ready

Dataset Overview

Total Images: 0
Total Annotations: 0

(a) 語彙・クラス一覧 (Models)

(b) 学習履歴・指標の確認 (Analytics)

Current Model

Model Path: ./yolov8s-world.pt

Classes: 11

Available Models

- backup_model_backup_20250929_003209
9/28/2025, 4:51:55 PM
model_backups/model_backup_20250929_003209.pt

Analytics Dashboard

Overview

Dataset

Performance

Images: 0

Labels: 0

Models: 0

Training Runs: 0

Recent Training Runs

(c) モデルの切替・管理

(d) モデル概要の確認

図 3.3: 提案 UI の反復ループ (3/4) : 語彙・履歴・モデル管理

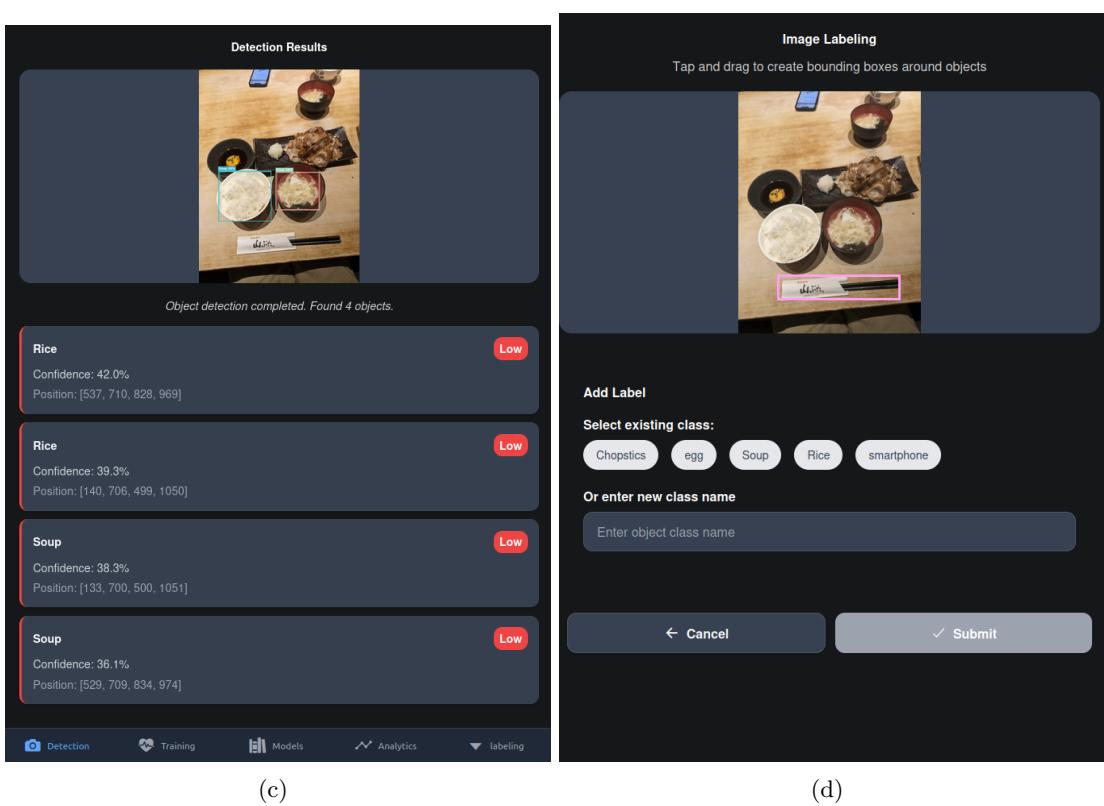
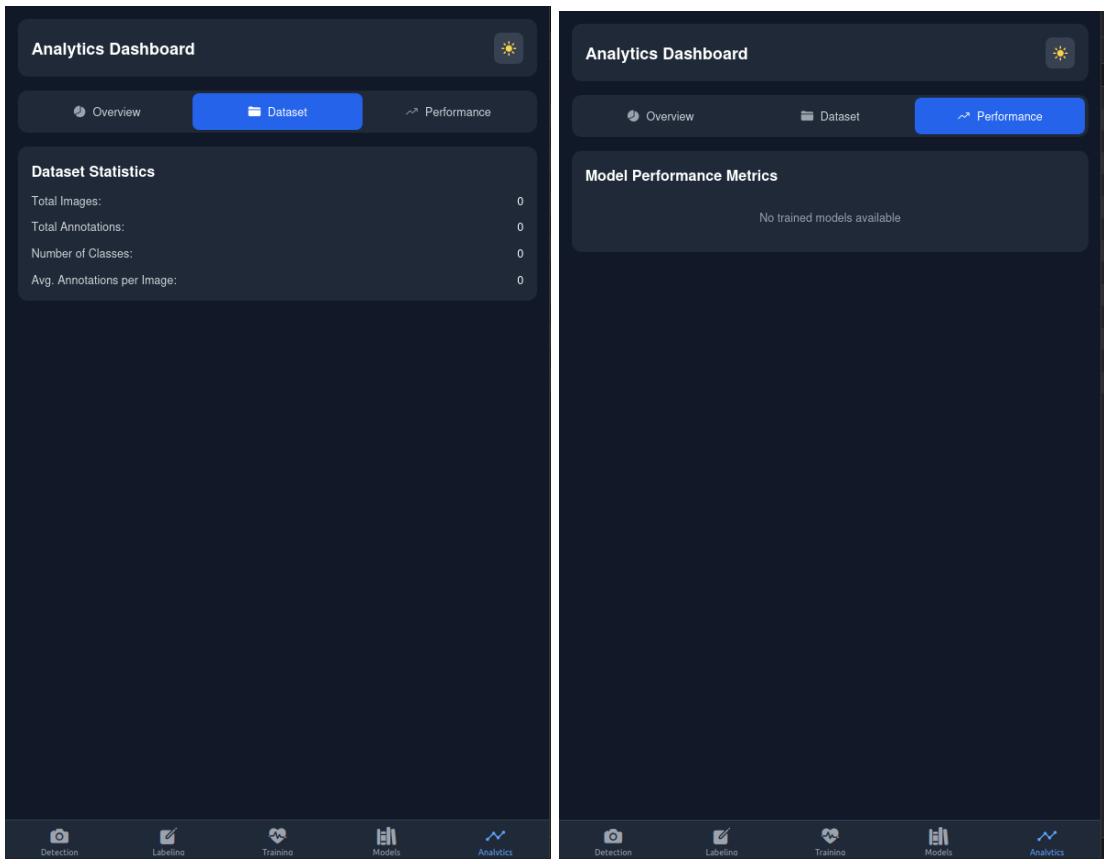


図 3.4: 提案 UI の反復ループ (4/4) : データ・性能の確認と再反復

3.5 スマホ最適化と制約

- **推論効率:** 事前語彙化によりテキストエンコーダを実行時から排除し、端末上のレイテンシを低減。
- **操作負荷の低減:** 撮影→検出→語彙追加→学習の短サイクルを UI で誘導し、少量データでも改善可能に。
- **制約:** 既定は CPU 学習であり大規模学習は長時間を要する。一方で CUDA 対応マシンでは設定により GPU 学習・推論へ切替可能であり、反復時間を短縮できる。現状は学習/検証分割を簡便化しており、厳密評価は今後の拡張で対応する。

以上より、**ユーザ主導の反復改善**（語彙設定→収集/ラベリング→学習→推論/評価→運用）を一つの UI に束ね、スマートフォンを中心とした現場適用に耐える**軽量な開放語彙検出運用**を実現する。

4

語彙登録のみでの検出成立性の検証

4.1 目的

本章では、**語彙登録 (Add New Detection Class)** のみで未検出の対象が検出可能になるかをスマートフォンから検証する。モデルはYOLO-World の**事前語彙化**運用であり、UI 上の語彙追加が POST /model/classes に対応、検出は POST /detect に対応する。対象例として smartphone クラスを用い、Home → Capture Image → Add New Detection Class → Detect の最短ループで成立性を確認する（図 3.1, 図 3.2 参照）。

4.2 操作フローと条件

1. Home でカメラ撮影またはギャラリーから画像選択（初回検出を実行し、対象が未検出であることを確認）。
2. 入力欄に新規クラス名（例:smartphone）を入力し、Add New Detection Class を押下（POST /model/classes）。
3. 直後に同一画像で再度検出（POST /detect）。語彙が反映され、対象が検出されるかを記録する。

語彙は custom_vocab.json に永続化され、アプリ再起動後も維持される。重複語彙は自動で除外され、空白のみの入力は無効化される。

4.3 評価項目

- **検出成立判定:** 語彙追加前後の検出有無（バウンディングボックスとクラス名の一一致）。
- **語彙反映時間:** POST /model/classes 送信から/detect 結果に反映されるまでの体感時間（秒）。
- **推論レイテンシ:** 画像 1 枚あたりの検出完了までの時間（UI 表示基準）。

4.4 ケーススタディ (smartphone)

スマートフォンの画像を対象に、初回は未検出であっても smartphone を語彙追加後に再検出すると検出成功するケースを確認した。UI 例は Add New Detection Class での登録画面（図 3.1）と、再検出での成立（図 3.2）を参照。

4.5 考察と制約

語彙登録のみで検出成立が得られる場合、追加学習なしで現場適合の初期効果が得られる。一方で、外観多様性が大きい対象やカメラ条件の変動が大きい場合は、語彙登録だけでは安定検出に至らない可能性がある。この際は次章の手動ラベリング+ファインチューニングに移行する。

5

手動ラベリングとファインチューニングによる検出成立性の検証

5.1 目的

語彙登録（4章）だけでは検出できなかった対象に対し、スマホからの手動ラベリングと短時間のファインチューニングで検出が成立するかを検証する。対象はUI上のManual Labeling (POST /labeling/submit) と Training (POST /training/start-async) を用いた最短反復で評価する。

5.2 前提と環境

フロントエンドはReact Native + Expo、バックエンドはFastAPI。学習は既定でCPUだが、環境設定によりGPUへ切替可能。収集した画像とYOLO形式ラベルは`training_data/images`・`training_data/labels`へ保存され、クラス一覧は`training_data/classes.txt`で管理される。学習設定`data.yaml`はAPIが自動生成する。

5.3 手順

1. Home で検出を実行し、目標対象が未検出であることを確認。
2. Manual Labeling で矩形とラベル名を付与し Submit (/labeling/submit)。これを少量ずつ（例:10～50枚）蓄積。
3. Training 画面から非同期学習を起動 (/training/start-async?epochs=E)。E は端末状況に応じて調整（例:20/50/100）。
4. /training/status で進捗を確認。完了時に最良重み best.pt が自動ロードされる。
5. Home に戻り同一/類似画像で再検出し、検出成立の有無を確認。

5.4 評価設計

- 検出成立率: 学習前後での検出有無の比較（正しくバウンディング・分類できた割合）。
- 必要ラベル数の目安: 初回の検出成立に必要だったラベル枚数の概算。
- 反復時間: ラベリング→学習→検証の1サイクル所要時間（端末体感）。
- 副作用の観察: 語彙衝突・誤検出の増減（類義語追加時など）。

5.5 実装上の要点

- **ラベル保存:** 送信データは YOLO 形式で保存され、クラスは `classes.txt` へ自動追記。新規クラスは `/model/classes` へ同期される。
- **学習設定:** `data.yaml` は API が自動生成。簡便化のため学習/検証は同一ディレクトリだが、将来的に分割を厳密化予定。
- **モデル反映:** 学習完了後、最良重みを自動ロード。Models タブで一覧・切替・バックアップ・簡易検証が可能。

5.6 実験結果

現時点ではデータセット未確定のため数値結果は未掲載。運用フローと評価指標のみ提示する。

5.7 期待と限界

語彙登録だけで立ち上がらない対象でも、**少量ラベル+短時間学習**で検出が成立することを期待する。一方で、撮影条件や外観多様性が大きい場合は追加データ収集と反復学習が必要。モバイル中心運用のため、**学習/推論時間と電力**の制約がある。

5.8 今後の改善

データ分割の厳密化、GPU 環境での高速化、半自動ラベリング支援、同義語正規化、学習履歴の可視化充実 (`/training/history`, `/training/metrics/{run_name}` の活用) を進める。

6

まとめ

本稿では、画像認識 AI を用いて食事の画像から残量測定を行った。Semantic Segmentation (意味的領域分割) モデルと Object Detection (物体検出) モデルを用いてそれを実現し、複数のモデルを使うことによる必要リソースの増加を抑えるためにユニオンモデルを提案した。そのユニオンモデルのうち、Semantic Segmentation (領域分割) の mIoU は最も高いもので 91.13%・物体検出の mIoU は最も高いもので 95.83% を達成した。本実験のデータセットは枚数が 241 枚と少ないため、測定用のデータセットでは精度が低く、汎化性能が落ちている。よって、汎化性能の向上のためにデータセットの増強の実施や対応クラスの増加を行うことが今後の課題である。加えて、推定手法として食器を球の一部に近似する球冠ベースと n 次関数に近似する関数ベースの手法を提案した。結果として、推定精度の最も高いものは球冠ベースであり、ご飯の RMSE は 7.6% を達成した。しかし、RMSE はまだまだ大きく、推定精度が良いとは言えない。その推定精度は領域分割精度・物体検出精度・推定手法に依存している。本実験の推定手法は凹凸の考慮が出来ず、食器の近似にも限界がある。よって、推定精度向上を達成するために推定手法の探索を行うことが今後の研究課題である。

謝辞

本研究に使用した画像はドリギー株式会社様から提供いただきました。本研究を進めるにあたり、終始熱心なご指導を頂いた孟林教授に深く感謝いたします。また、本研究において手助けをしていただいた石橋さんや研究室の皆様に感謝の念が絶えません。本当にありがとうございました。

研究業績

査読付き国際学会

1. **Haruhiro Takahashi**, Ryuto Ishibashi, Hayata Kaneko, and Lin Meng, “Leftover Food Measurement using Deep Learning Based Semantic Segmentation,” The 6th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2024), Aug. 2024. (in Kusatsu, Japan)
2. Ishibashi, Ryuto, **Haruhiro Takahashi**, and Lin Meng. ”ViT-Based Hybrid Segmentation for Leftover Food Detection.” The 6th International Conference on Industrial Artificial Intelligence (IAI2024). Aug. 2024. (in Shenyang, China)

シンポジウム

1. **Haruhiko Takahashi**, “Leftover Food Measurement using Segmentation and Detection”, The 21th English Presentation Competition in Ritsumeikan University (EPCR2024) ,Nov. 2024 (in Kusatsu, Japan)

参考文献

- [1] K. Cheng, Z. Xu, X. Wang, J. Dai, Y. Qiao, M. Tang, and H. Bai, “YOLO-World: Real-Time Open-Vocabulary Object Detection,” arXiv:2401.17270, 2024. <https://arxiv.org/abs/2401.17270>

付録 A

実験結果

A.1 領域検出精度の実験結果一覧

表 A.1、A.2 は、領域検出精度に関する実験の結果一覧であり、前者はユニオンモデルでの結果、後者はユニオンモデル 2 での結果である。

A.2 測定の実験結果一覧

表 A.3、A.4 は、測定に関する実験の結果一覧であり、前者はユニオンモデルでの結果、後者はユニオンモデル 2 での結果である。表中には球冠ベースと関数ベースでの推定手法のご飯・みそ汁に対する RMSE を掲載している。

図 A.1,A.2 はユニオンモデルでの測定の結果画像から抜粋したものであり、前者は球冠ベース・後者は関数ベースでの結果である。図 A.3,A.4 はユニオンモデル 2 での測定の結果画像から抜粋したものであり、前者は球冠ベース・後者は関数ベースでの結果である。一番上の数字は実測値であり、画像の左横にある文字は使用したモデル、画像の下にある数字はそのモデルでの推定値である。

Backbone	Method		Seg mIoU[%]	Det		FLOPs [G]	Params [M]	Latency [ms]
	Seg	Det		mAP[%]	mIoU[%]			
MB_v3_s	U-Net	YOLOX	88.11	100	93.8	0.84	3.67	8.38
MB_v3_s	DL3+	YOLOX	89.44	100	93.8	4.88	8.97	8.27
MB_v3_s	DL3	YOLOX	75.97	100	93.8	0.85	8.27	8.23
MB_v3_l	U-Net	YOLOX	90.36	100	95.17	1.62	8.44	9.03
MB_v3_l	DL3+	YOLOX	90.59	100	95.17	5.72	16.30	9.09
MB_v3_l	DL3	YOLOX	78.35	100	95.17	1.69	15.60	8.76
EF_b0	U-Net	YOLOX	90.12	100	95.02	2.40	11.06	10.10
EF_b0	DL3+	YOLOX	91.13	100	95.02	6.58	21.14	10.03
EF_b0	DL3	YOLOX	77.01	100	95.02	2.55	20.44	9.99
EF_b1	U-Net	YOLOX	90.4	100	94.43	3.18	16.08	12.19
EF_b1	DL3+	YOLOX	88.89	100	94.43	7.36	26.15	12.05
EF_b1	DL3	YOLOX	75.15	100	94.43	3.33	25.45	11.89
EF_b2	U-Net	YOLOX	90.31	99.68	95.06	3.57	18.73	11.90
EF_b2	DL3+	YOLOX	90.82	99.68	95.06	7.78	29.67	12.35
EF_b2	DL3	YOLOX	74.84	99.68	95.06	3.75	28.97	11.98
EF_b3	U-Net	YOLOX	90.46	100	95.63	4.88	25.03	13.12
EF_b3	DL3+	YOLOX	89.23	100	95.63	9.11	36.84	12.89
EF_b3	DL3	YOLOX	75.49	100	95.63	5.08	36.14	13.05
EF_b4	U-Net	YOLOX	90.65	100	95.47	7.18	39.41	15.05
EF_b4	DL3+	YOLOX	90.99	100	95.47	11.46	52.95	14.99
EF_b4	DL3	YOLOX	75.6	100	95.47	7.43	52.25	14.90
EF_b5	U-Net	YOLOX	90.29	100	95.68	10.77	61.71	16.99
EF_b5	DL3+	YOLOX	88.63	100	95.68	15.11	77.00	16.97
EF_b5	DL3	YOLOX	72.83	100	95.68	11.08	76.30	16.47
EF_b6	U-Net	YOLOX	89.64	100	95.79	14.97	87.32	18.80
EF_b6	DL3+	YOLOX	88.88	100	95.79	19.36	104.34	18.58
EF_b6	DL3	YOLOX	71.31	100	95.79	15.33	103.64	19.01
EF_b7	U-Net	YOLOX	90.78	100	95.8	22.47	134.32	21.89
EF_b7	DL3+	YOLOX	90.31	100	95.8	26.91	153.08	21.49
EF_b7	DL3	YOLOX	73.87	100	95.8	22.88	152.37	21.16
R_18	U-Net	YOLOX	90.02	100	94.4	4.36	13.70	8.05
R_18	DL3+	YOLOX	88.94	100	94.4	8.25	18.24	6.68
R_18	DL3	YOLOX	72.85	100	94.4	4.21	17.54	6.58
R_34	U-Net	YOLOX	88.35	99.92	95.32	6.91	23.89	7.71
R_34	DL3+	YOLOX	89.29	99.92	95.32	10.80	28.43	7.75
R_34	DL3	YOLOX	73.32	99.92	95.32	6.77	27.72	7.61
R_50	U-Net	YOLOX	90.5	99.3	95.81	8.83	34.87	8.97
R_50	DL3+	YOLOX	90.64	99.3	95.81	12.61	48.87	8.84
R_50	DL3	YOLOX	76.41	99.3	95.81	8.55	48.16	8.55
R_101	U-Net	YOLOX	90.38	100	95.5	12.57	53.86	12.49
R_101	DL3+	YOLOX	90.75	100	95.5	16.35	67.86	12.21
R_101	DL3	YOLOX	75.72	100	95.5	12.28	67.15	12.24
R_152	U-Net	YOLOX	89	100	95.83	16.30	69.51	16.06
R_152	DL3+	YOLOX	90.53	100	95.83	20.08	83.51	15.88
R_152	DL3	YOLOX	74.87	100	95.83	16.02	82.79	15.43

表 A.1: 学習結果(ユニオンモデル)

図 A.1: 球冠ベースの結果画像(ユニオンモデル)

図 A.2: 関数ベースの結果画像(ユニオンモデル)

Backbone	Method		Seg mIoU[%]	Det		FLOPs [G]	Params [M]	Latency [ms]
	Seg	Det		mAP[%]	mIoU[%]			
MB_v3_s	U-Net	YOLOX	90	99.57	91.05	0.84	3.67	8.38
MB_v3_s	DL3+	YOLOX	90.5	99.3	93.15	4.88	8.97	8.27
MB_v3_s	DL3	YOLOX	83.35	100	92.67	0.85	8.27	8.23
MB_v3_l	U-Net	YOLOX	90.42	99.96	93.06	1.62	8.44	9.03
MB_v3_l	DL3+	YOLOX	90.91	100	94.61	5.72	16.30	9.09
MB_v3_l	DL3	YOLOX	84.6	100	95.08	1.69	15.60	8.76
EF_b0	U-Net	YOLOX	90.82	99.3	93.82	2.40	11.06	10.10
EF_b0	DL3+	YOLOX	91.2	100	94.45	6.58	21.14	10.03
EF_b0	DL3	YOLOX	85.05	99.29	94.71	2.55	20.44	9.99
EF_b1	U-Net	YOLOX	90.9	99.3	93.65	3.18	16.08	12.19
EF_b1	DL3+	YOLOX	91.17	99.3	95.14	7.36	26.15	12.05
EF_b1	DL3	YOLOX	80.26	100	94.51	3.33	25.45	11.89
EF_b2	U-Net	YOLOX	91.03	100	93.32	3.57	18.73	11.90
EF_b2	DL3+	YOLOX	91.33	100	94.88	7.78	29.67	12.35
EF_b2	DL3	YOLOX	83.87	100	94.54	3.75	28.97	11.98
EF_b3	U-Net	YOLOX	91.21	100	93.03	4.88	25.03	13.12
EF_b3	DL3+	YOLOX	91.29	100	94.29	9.11	36.84	12.89
EF_b3	DL3	YOLOX	84.73	100	94.98	5.08	36.14	13.05
EF_b4	U-Net	YOLOX	91.03	100	95.04	7.18	39.41	15.05
EF_b4	DL3+	YOLOX	91.33	100	94.84	11.46	52.95	14.99
EF_b4	DL3	YOLOX	84.1	100	95.67	7.43	52.25	14.90
EF_b5	U-Net	YOLOX	91.18	100	94.01	10.77	61.71	16.99
EF_b5	DL3+	YOLOX	83.63	100	95.53	15.11	77.00	16.97
EF_b5	DL3	YOLOX	48.4	99.99	94.16	11.08	76.30	16.47
EF_b6	U-Net	YOLOX	91.32	100	94.62	14.97	87.32	18.80
EF_b6	DL3+	YOLOX	91.27	100	95.33	19.36	104.34	18.58
EF_b6	DL3	YOLOX	82.89	100	94.68	15.33	103.64	19.01
EF_b7	U-Net	YOLOX	91.24	100	94.62	22.47	134.32	21.89
EF_b7	DL3+	YOLOX	91.27	99.3	95.59	26.91	153.08	21.49
EF_b7	DL3	YOLOX	84.47	100	95.54	22.88	152.37	21.16
R_18	U-Net	YOLOX	90.42	100	94.5	4.36	13.70	8.05
R_18	DL3+	YOLOX	91.08	100	95.52	8.25	18.24	6.68
R_18	DL3	YOLOX	84.13	100	95.57	4.21	17.54	6.58
R_34	U-Net	YOLOX	90.28	97.79	93.47	6.91	23.89	7.71
R_34	DL3+	YOLOX	91.01	100	94	10.80	28.43	7.75
R_34	DL3	YOLOX	84.31	100	95.37	6.77	27.72	7.61
R_50	U-Net	YOLOX	90.62	99.3	94.17	8.83	34.87	8.97
R_50	DL3+	YOLOX	91.03	99.98	94.61	12.61	48.87	8.84
R_50	DL3	YOLOX	83.83	99.3	95.61	8.55	48.16	8.55
R_101	U-Net	YOLOX	90.28	99.29	94.01	12.57	53.86	12.49
R_101	DL3+	YOLOX	91.18	99.99	94.97	16.35	67.86	12.21
R_101	DL3	YOLOX	66.37	100	95.59	12.28	67.15	12.24
R_152	U-Net	YOLOX	90.31	100	94.66	16.30	69.51	16.06
R_152	DL3+	YOLOX	91.01	99.96	92.84	20.08	83.51	15.88
R_152	DL3	YOLOX	44.14	99.3	95.12	16.02	82.79	15.43

表 A.2: 学習結果(ユニオンモデル 2)

図 A.3: 球冠ベースの結果画像(ユニオンモデル 2)

図 A.4: 関数ベースの結果画像(ユニオンモデル 2)

Backbone	Method		Seg mIoU[%]	Det mIoU[%]	球冠ベース		関数ベース	
	Seg	Det			ご飯 [%]	みそ汁 [%]	ご飯 [%]	みそ汁 [%]
MB_v3_s	Unet	YOLOX	88.11	93.8	9.10	29.62	9.90	25.30
MB_v3_s	DLv3+	YOLOX	89.44	93.8	10.68	31.99	10.99	28.90
MB_v3_s	DLv3	YOLOX	75.97	93.8	13.86	29.01	14.28	29.24
MB_v3_l	Unet	YOLOX	90.36	95.17	11.64	22.76	12.23	15.72
MB_v3_l	DLv3+	YOLOX	90.59	95.17	12.25	25.43	12.86	19.50
MB_v3_l	DLv3	YOLOX	78.35	95.17	15.91	34.29	16.54	31.85
EF_b0	Unet	YOLOX	90.12	95.02	15.49	26.73	16.01	23.31
EF_b0	DLv3+	YOLOX	91.13	95.02	12.21	31.78	12.64	23.56
EF_b0	DLv3	YOLOX	77.01	95.02	17.13	36.70	17.22	33.37
EF_b1	Unet	YOLOX	90.4	94.43	12.13	21.03	12.78	14.16
EF_b1	DLv3+	YOLOX	88.89	94.43	9.27	24.33	9.88	17.41
EF_b1	DLv3	YOLOX	75.15	94.43	17.77	25.15	17.22	25.74
EF_b2	Unet	YOLOX	90.31	95.06	10.90	30.35	11.69	26.04
EF_b2	DLv3+	YOLOX	90.82	95.06	11.41	24.65	12.14	21.05
EF_b2	DLv3	YOLOX	74.84	95.06	17.03	31.27	17.31	34.93
EF_b3	Unet	YOLOX	90.46	95.63	15.30	29.15	15.72	20.81
EF_b3	DLv3+	YOLOX	89.23	95.63	18.48	33.50	18.80	27.92
EF_b3	DLv3	YOLOX	75.49	95.63	21.46	39.17	21.75	39.27
EF_b4	Unet	YOLOX	90.65	95.47	12.66	24.18	13.16	16.97
EF_b4	DLv3+	YOLOX	90.99	95.47	16.61	33.09	17.04	26.58
EF_b4	DLv3	YOLOX	75.60	95.47	15.70	35.10	16.49	32.87
EF_b5	Unet	YOLOX	90.29	95.68	11.45	20.84	12.26	15.12
EF_b5	DLv3+	YOLOX	88.63	95.68	14.67	27.32	15.61	20.21
EF_b5	DLv3	YOLOX	72.83	95.68	24.84	36.96	25.99	35.17
EF_b6	Unet	YOLOX	89.64	95.79	14.03	18.70	14.57	12.78
EF_b6	DLv3+	YOLOX	88.88	95.79	15.52	34.16	16.00	29.81
EF_b6	DLv3	YOLOX	71.31	95.79	23.32	40.29	24.04	37.70
EF_b7	Unet	YOLOX	90.78	95.8	12.25	21.49	12.84	14.94
EF_b7	DLv3+	YOLOX	90.31	95.8	13.20	24.73	14.10	19.22
EF_b7	DLv3	YOLOX	73.87	95.8	22.18	29.01	22.46	29.41
R_18	Unet	YOLOX	90.02	94.4	7.56	20.82	8.44	17.01
R_18	DLv3+	YOLOX	88.94	94.4	10.08	20.98	10.61	19.84
R_18	DLv3	YOLOX	72.85	94.4	19.46	31.71	19.47	32.32
R_34	Unet	YOLOX	88.35	95.32	14.08	20.84	14.85	21.84
R_34	DLv3+	YOLOX	89.29	95.32	13.56	19.05	14.09	15.33
R_34	DLv3	YOLOX	73.32	95.32	14.53	28.19	14.82	31.61
R_50	Unet	YOLOX	90.5	95.81	11.70	23.93	12.31	19.46
R_50	DLv3+	YOLOX	90.64	95.81	15.67	31.81	16.04	27.88
R_50	DLv3	YOLOX	76.41	95.81	17.29	43.98	17.21	44.58
R_101	Unet	YOLOX	90.38	95.5	14.20	24.99	14.69	17.76
R_101	DLv3+	YOLOX	90.75	95.5	13.05	38.83	13.63	33.50
R_101	DLv3	YOLOX	75.72	95.5	28.97	35.67	28.65	31.85
R_152	Unet	YOLOX	89.0	95.83	10.99	26.55	11.66	20.49
R_152	DLv3+	YOLOX	90.53	95.83	14.15	30.17	14.68	26.74
R_152	DLv3	YOLOX	74.87	95.83	18.07	40.91	17.64	38.12

表 A.3: 測定結果(ユニオンモデル)

Backbone	Method		Seg mIoU[%]	Det mIoU[%]	球冠ベース		関数ベース	
	Seg	Det			ご飯 [%]	みそ汁 [%]	ご飯 [%]	みそ汁 [%]
MB_v3_s	Unet	YOLOX	90	91.05	10.81	27.08	11.36	20.92
MB_v3_s	DLv3+	YOLOX	90.5	93.15	11.25	26.80	11.73	19.86
MB_v3_s	DLv3	YOLOX	83.35	92.67	9.47	29.20	10.03	23.41
MB_v3_l	Unet	YOLOX	90.42	93.06	10.74	22.99	11.43	21.45
MB_v3_l	DLv3+	YOLOX	90.91	94.61	12.46	25.62	13.03	21.18
MB_v3_l	DLv3	YOLOX	84.6	95.08	10.71	24.49	11.54	20.10
EF_b0	Unet	YOLOX	90.82	93.82	12.34	21.76	12.94	14.87
EF_b0	DLv3+	YOLOX	91.2	94.45	11.69	26.85	12.24	20.49
EF_b0	DLv3	YOLOX	85.05	94.71	12.22	21.81	12.88	19.55
EF_b1	Unet	YOLOX	90.9	93.65	13.24	15.31	13.72	8.54
EF_b1	DLv3+	YOLOX	91.17	95.14	14.59	30.15	14.93	27.28
EF_b1	DLv3	YOLOX	80.26	94.51	9.89	31.78	10.20	31.04
EF_b2	Unet	YOLOX	91.03	93.32	13.11	18.60	13.45	13.36
EF_b2	DLv3+	YOLOX	91.33	94.88	11.06	21.02	11.66	16.69
EF_b2	DLv3	YOLOX	83.87	94.54	9.48	26.36	10.50	20.11
EF_b3	Unet	YOLOX	91.21	94.62	13.83	27.33	14.20	23.50
EF_b3	DLv3+	YOLOX	91.29	95.59	13.46	33.36	14.10	30.63
EF_b3	DLv3	YOLOX	84.73	95.54	11.50	21.67	12.37	16.30
EF_b4	Unet	YOLOX	91.03	95.04	11.82	16.34	12.42	10.09
EF_b4	DLv3+	YOLOX	91.33	94.84	11.93	26.62	12.49	21.19
EF_b4	DLv3	YOLOX	84.1	95.67	9.92	23.36	10.81	17.09
EF_b5	Unet	YOLOX	91.18	94.01	12.17	30.94	12.71	29.03
EF_b5	DLv3+	YOLOX	83.63	95.53	37.16	29.61	37.34	26.00
EF_b5	DLv3	YOLOX	48.4	94.16	51.73	30.36	51.73	24.36
EF_b6	Unet	YOLOX	91.32	94.62	11.87	26.23	12.30	21.53
EF_b6	DLv3+	YOLOX	91.27	95.33	13.87	18.94	14.49	12.76
EF_b6	DLv3	YOLOX	82.89	94.68	11.33	19.14	12.18	13.36
EF_b7	Unet	YOLOX	91.24	94.62	13.83	27.33	14.20	23.50
EF_b7	DLv3+	YOLOX	91.27	95.59	13.46	33.36	14.10	30.63
EF_b7	DLv3	YOLOX	84.47	95.54	11.50	21.67	12.37	16.30
R_18	Unet	YOLOX	90.42	94.5	8.32	15.64	9.20	9.30
R_18	DLv3+	YOLOX	91.08	95.52	13.25	35.07	13.77	34.58
R_18	DLv3	YOLOX	84.13	95.57	6.24	25.81	7.06	19.73
R_34	Unet	YOLOX	90.28	93.47	11.27	31.01	12.01	29.26
R_34	DLv3+	YOLOX	91.01	94	11.41	31.95	12.18	29.50
R_34	DLv3	YOLOX	84.31	95.37	8.07	29.01	8.71	24.48
R_50	Unet	YOLOX	90.62	94.17	10.27	19.11	11.08	12.90
R_50	DLv3+	YOLOX	91.03	94.61	13.88	14.95	14.47	9.79
R_50	DLv3	YOLOX	83.83	95.61	11.31	34.77	11.98	32.03
R_101	Unet	YOLOX	90.28	94.01	11.36	27.72	11.92	24.35
R_101	DLv3+	YOLOX	91.18	94.97	13.00	31.78	13.53	29.60
R_101	DLv3	YOLOX	66.37	95.59	18.68	34.50	19.45	32.05
R_152	Unet	YOLOX	90.31	94.66	11.38	31.54	12.05	29.57
R_152	DLv3+	YOLOX	91.01	92.84	13.85	38.86	14.18	36.40
R_152	DLv3	YOLOX	44.14	95.12	51.73	45.33	51.73	45.15

表 A.4: 測定結果(ユニオンモデル2)