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を中核とする検出・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内でのデータ収集・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 スマートフォンを用いたユーザ主導実験設計 **11**
4.1 目的とモチベーション 11
4.2 対象ユーザと使用環境 11
4.3 タスク定義 11
4.4 プロトコル（手順） 11
4.5 評価指標 12
4.5.1 機能的指標 12
4.5.2 運用指標 12
4.5.3 主観評価 12
4.6 ログ・記録 12
4.7 倫理配慮とプライバシ 12
4.8 成功基準と終了条件 12
4.9 想定リスクと緩和 12
4.10 結果の反映計画 12

5 領域検出精度に関する実験 **13**
5.1 目的 13
5.2 データセット 13
5.3 実験条件 13
5.4 評価指標 14
5.4.1 Tversky loss 14
5.4.2 IoU 14
5.4.3 mAP 14
5.4.4 YOLOX loss 15
5.5 実験結果 15
5.6 考察 16

6 残量測定に関する実験 **18**
6.1 目的 18
6.2 実験条件 18
6.2.1 推定用データセット 18
6.2.2 評価指標 18
6.3 実験結果 18
6.4 考察 19

7 まとめ **20**

A 実験結果 **24**
A.1 領域検出精度の実験結果一覧 24
A.2 測定の実験結果一覧 24

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

5.1 意味的領域分割タスク用のラベル 13
5.2 物体検出タスク用のラベル 13
5.3 領域の分割の概要 14
5.4 P(r) 14
5.5 Semantic Segmentation (領域分割) のバブルグラフ (ユニオンモデル) 15
5.6 Object Detection (物体検出) のバブルグラフ (ユニオンモデル) 15
5.7 領域分割のバブルグラフ (ユニオンモデル 2) 16
5.8 物体検出のバブルグラフ (ユニオンモデル 2) 16

6.1 画像とその数値 [%] 18
6.2 球冠ベース (ご飯) のバブルグラフ 19
6.3 球冠ベース (みそ汁) のバブルグラフ 19
6.4 関数ベース (ご飯) のバブルグラフ 19
6.5 関数ベース (みそ汁) のバブルグラフ 19

A.1 球冠ベースの結果画像 (ユニオンモデル) 25
A.2 関数ベースの結果画像 (ユニオンモデル) 25
A.3 球冠ベースの結果画像 (ユニオンモデル 2) 26
A.4 関数ベースの結果画像 (ユニオンモデル 2) 26

表目次

5.1 学習・推論条件および実験環境 13
5.2 学習結果の抜粋 (ユニオンモデル) 16
5.3 学習結果の抜粋 (ユニオンモデル 2) 16

6.1 測定結果の抜粋 (ユニオンモデル) 19

A.1 学習結果 (ユニオンモデル) 25
A.2 学習結果 (ユニオンモデル 2) 26
A.3 測定結果 (ユニオンモデル) 27
A.4 測定結果 (ユニオンモデル 2) 28

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 章では実装の詳細（フロントエンド、バックエンド、学習基盤、データ管理、可視化）を述べ、5 章でケーススタディと評価（ユーザ操作性、学習・推論特性、改善サイクルの有効性）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）：

$$X' \;=\; X \odot \mathrm{Sigmoid}\big(\max\big(\mathrm{Conv}(X, W), \dim = 1\big)\big)\,, \tag{2.1}$$

ここで X は画像特徴，W はテキスト埋め込み由来の畳み込み重み，$\odot$ 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 を単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

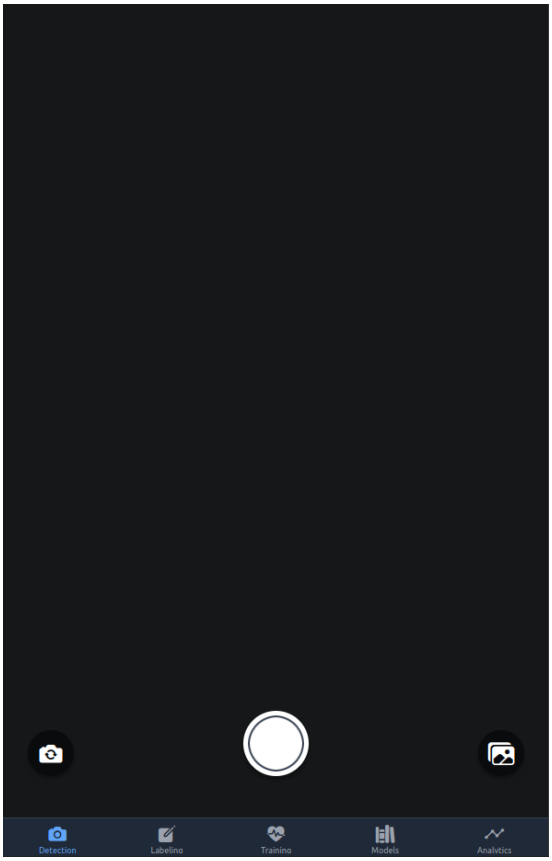
YOLO-World は視覚 – 言語モデリングにより、**ユーザ定義語彙**での開放語彙検出を実現する。`POST /model/classes` で登録した語彙は `custom_vocab.json` に永続化され、モデルのクラス埋め込みへ反映される。推論は`/detect` で実行し、バウンディングボックス・クラス・スコアと描画済み画像を返す。**事前語彙化**により、推論時はオフラインで固定した語彙埋め込みを用い、モバイルでも実用的なレイテンシを確保する。

3.3 データ収集・ラベリング・学習

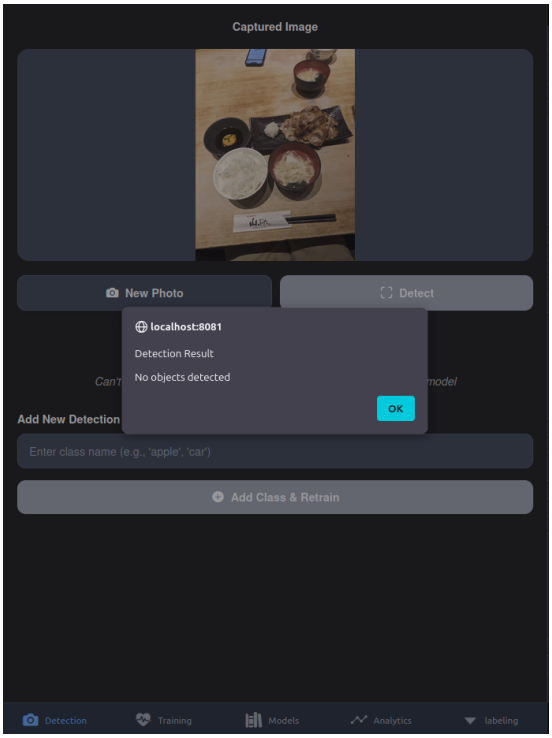
Labeling タブで作成したアノテーションは YOLO 形式で保存し（`training_data/` 配下）、`/training/start` または`/training/start-async` で微調整を起動する。**既定は CPU 実行**だが、**CUDA 対応マシンでは設定により GPU（device=‘cuda’）で学習・推論が可能**である。完了時には `best.pt` を自動ロードし、`/training/status` で進捗を可視化する。`/models/*`群でモデル一覧・切替・バックアップ・検証が可能で、`/training/history` と`/training/metrics/{run_name}`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

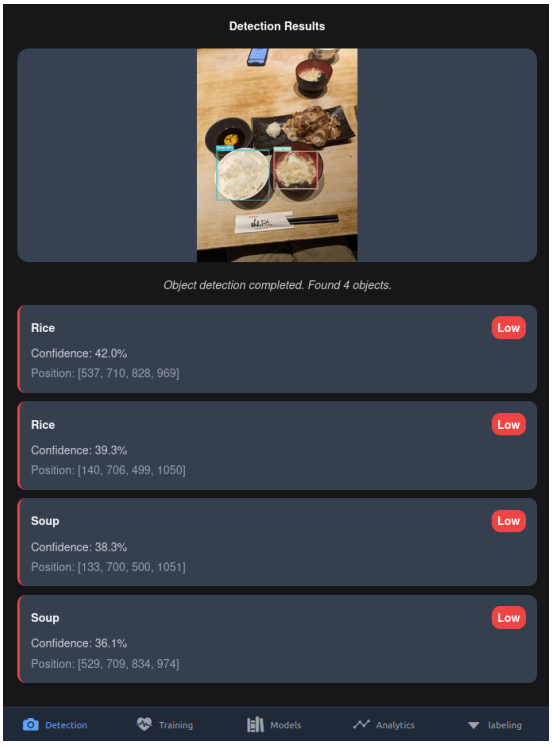
本節では、スマートフォン想定の**最短反復ループ**（撮影→検出→語彙追加→再検出→ラベリング→学習→評価）を画面遷移で示す。各ページに 4 枚ずつ配置する。



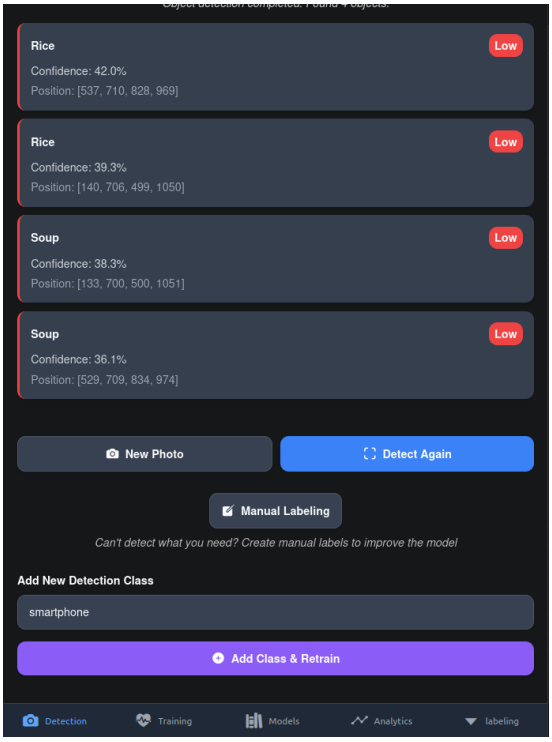
(a) 撮影/選択（Detection）



(b) 初回検出（未学習）

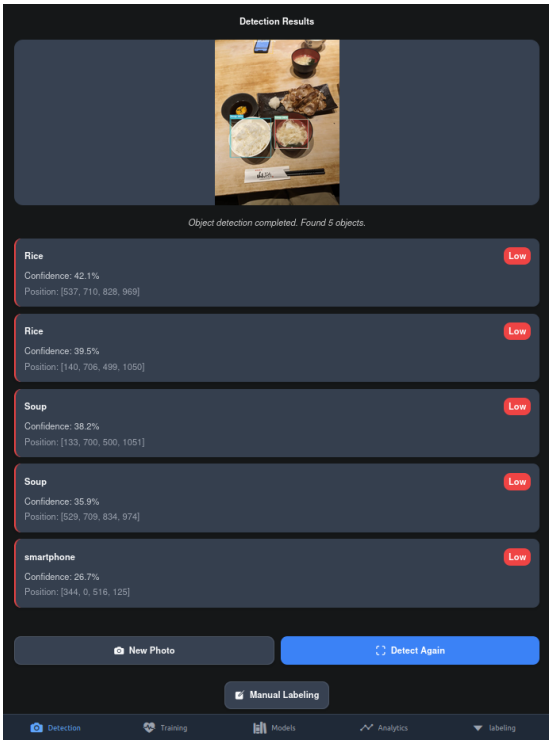


(c) クラス追加後の検出（既存クラス）

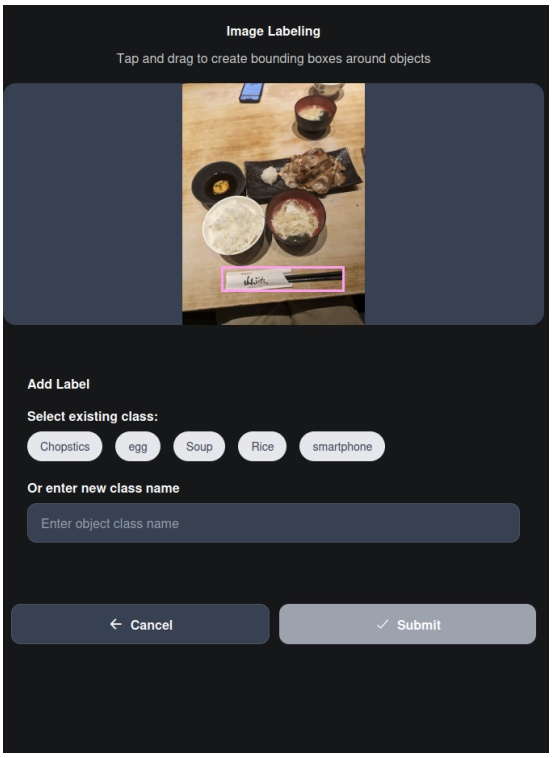


(d) スマートフォンクラスの追加（Models）

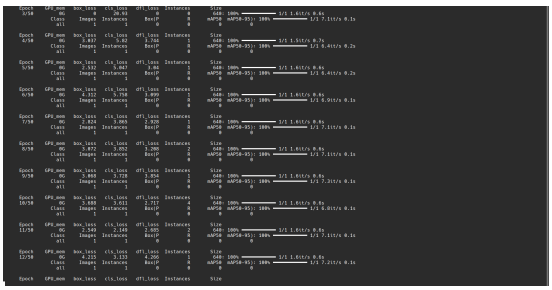
図 3.1: 提案 UI の反復ループ（1/4）：撮影→初回検出→語彙確認/追加



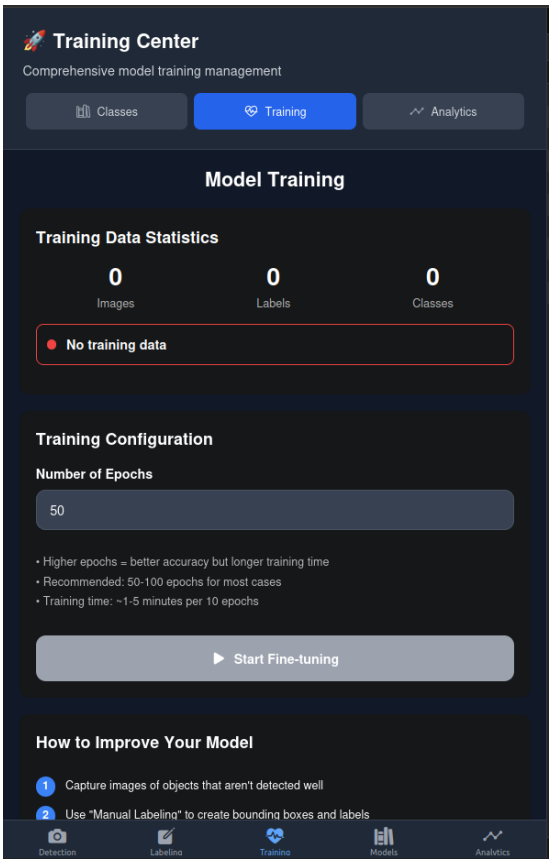
(a) smartphone クラスで検出成功



(b) マニュアルラベリング

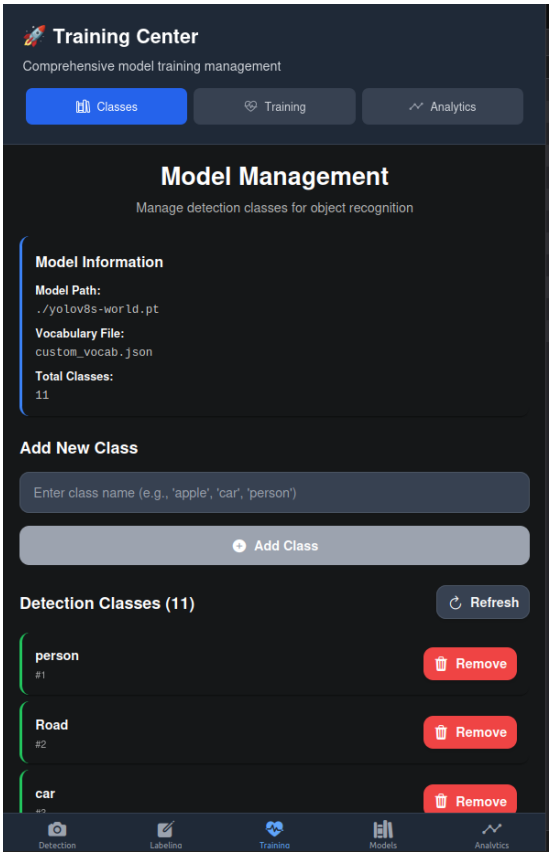


(c) ファインチューニング開始

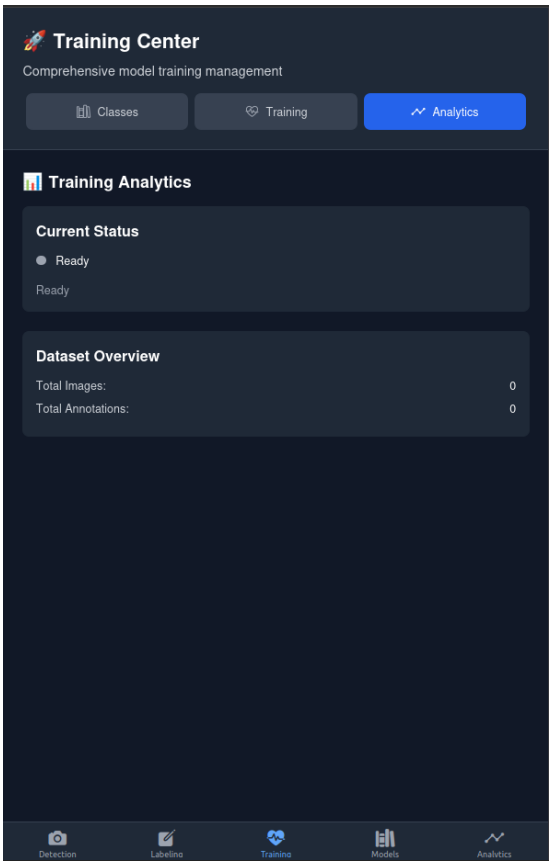


(d) 学習の起動（Training）

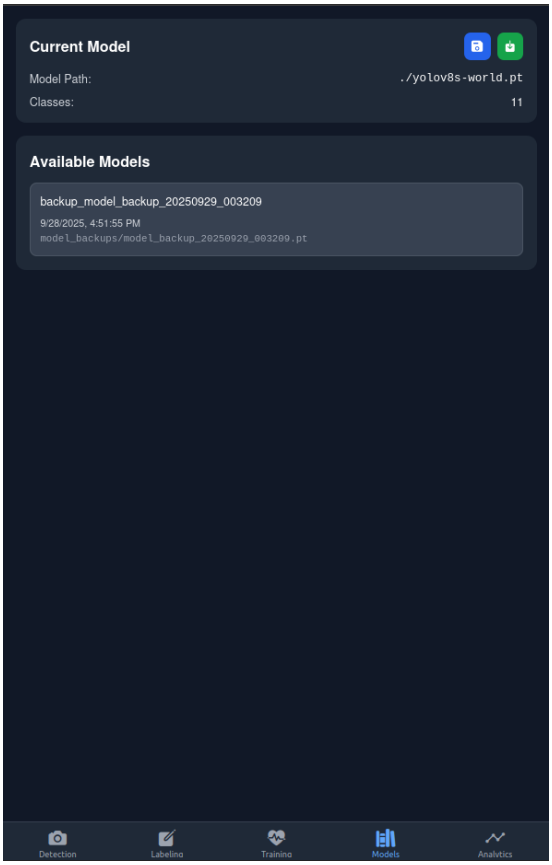
図 3.2: 提案 UI の反復ループ（2/4）：再検出→ラベリング→学習起動



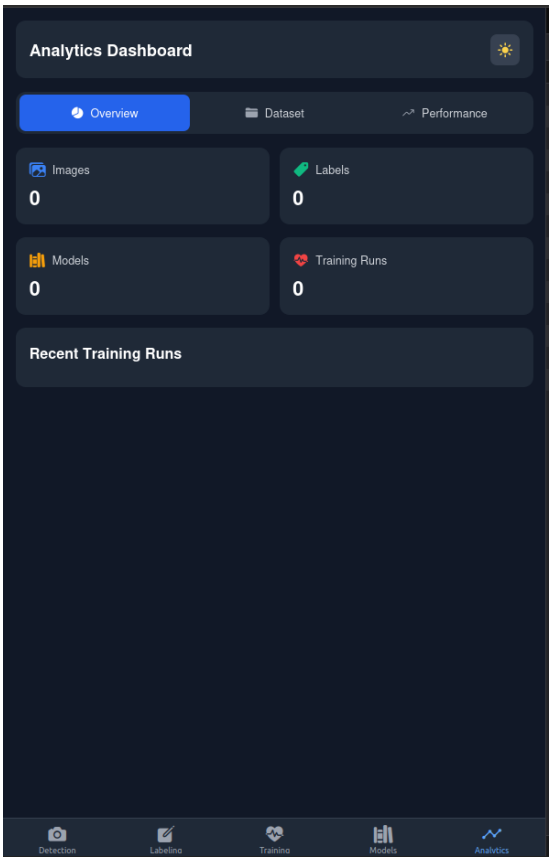
(a) 語彙・クラス一覧（Models）



(b) 学習履歴・指標の確認（Analytics）

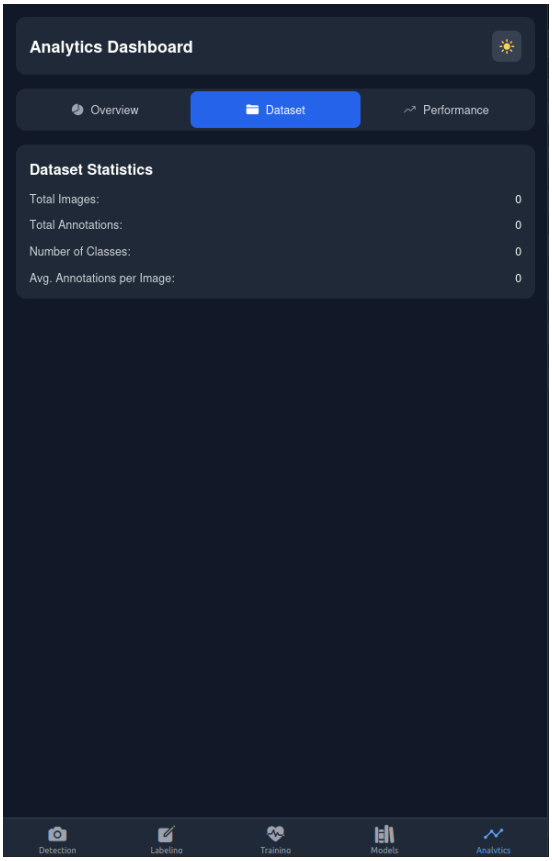


(c) モデルの切替・管理

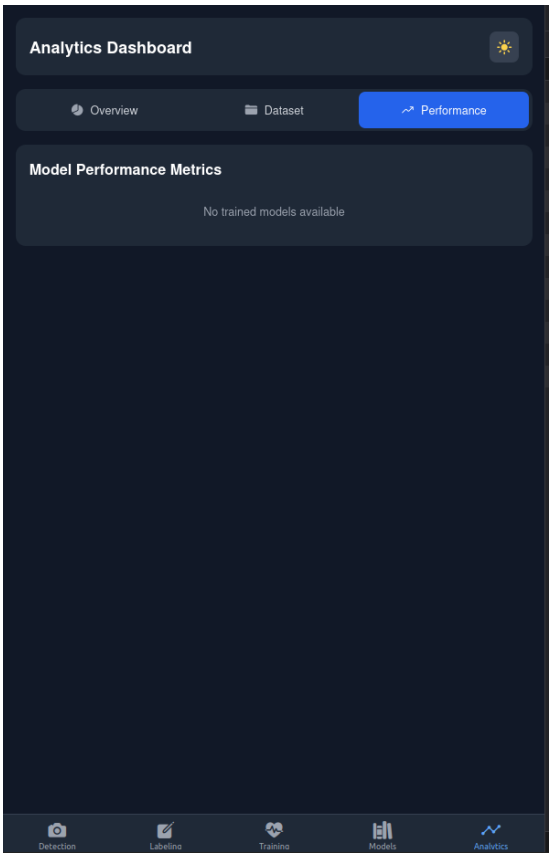


(d) モデル概要の確認

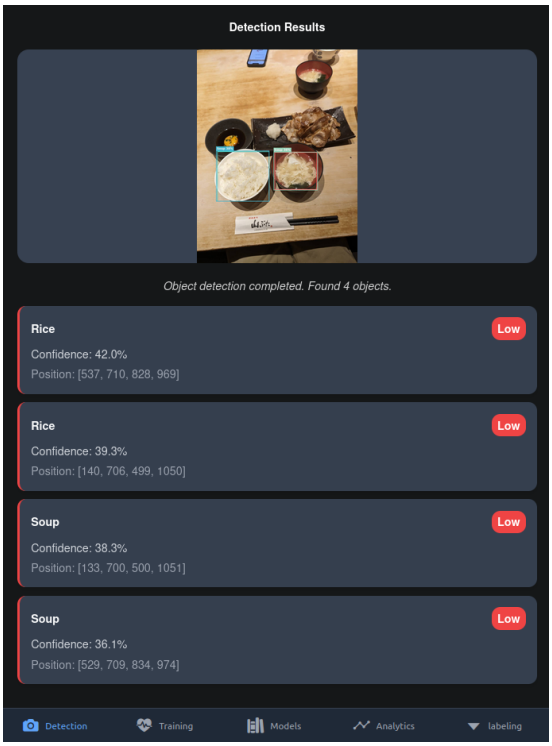
図 3.3: 提案 UI の反復ループ（3/4）：語彙・履歴・モデル管理



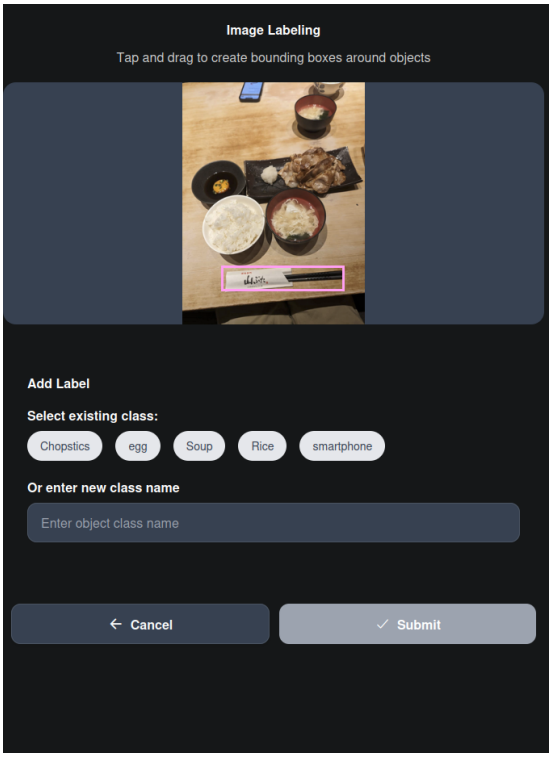
(a) データセット確認



(b) 性能比較・推移の確認



(c)



(d)

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

スマートフォンを用いたユーザ主導実験設計

4.1 目的とモチベーション

本研究の実験は、スマートフォンのみで画像取得から語彙設定・ラベリング・学習・推論までを完結できるかを検証する。一般ユーザが自作モデルを短時間で構築・改善できることを主たるモチベーションとし、操作の単純さと反復速度を重視する。

4.2 対象ユーザと使用環境

対象は機械学習の専門知識を持たない一般ユーザとし、端末は Android/iOS の実機を想定する。通信は Wi-Fi 環境を基本とし、バックエンド API はローカル同一 LAN またはトンネリングを用いて接続する。UI は Expo アプリ（タブ: Detection / Labeling / Training / Models / Analytics）を使用する。

4.3 タスク定義

- T1: スマホで対象物（例: 食器/料理）の写真を撮影またはギャラリーから選択する。
- T2: Labeling タブでバウンディングボックスとラベルを付与し送信する（`/labeling/submit`）。
- T3: 語彙（検出クラス）を`/model/classes` で登録・更新する（必要なら追加）。
- T4: Training タブから学習を起動（同期または非同期）。完了後にモデル自動ロードを確認する。
- T5: Detection タブで推論し、語彙が正しく反映され検出が成立するか確認する。

4.4 プロトコル（手順）

1. 初期語彙を空または最小集合で開始し、Detection タブで現状の検出挙動を確認する。
2. 必要語彙を追加（`POST /model/classes`）し、Labeling タブで 10〜30 枚程度を目安にアノテーションを実施する。
3. `/training/start-async` で学習を開始し、`/training/status` で進捗を監視する。完了後、自動ロードされたモデルで推論を再確認する。
4. 検出が不十分な語彙があればデータを追加収集・再ラベリングし、学習・評価を反復する（最大 3 サイクル程度）。

4.5 評価指標

4.5.1 機能的指標

- 検出成立率: 目標語彙に対し、正しくバウンディングされクラスが一致した割合。
- 語彙反映時間: 語彙登録から推論結果への反映までの所要時間。
- 推論レイテンシ: 端末での 1 画像あたりの表示までの体感時間（秒）。

4.5.2 運用指標

- 設定完了時間: 初回起動から「最初の正しい検出」達成までの時間。
- 反復コスト: 1 サイクル（収集→ラベル→学習→検証）に要する平均時間。

4.5.3 主観評価

- 使いやすさ（SUS の簡易版）: 操作の迷い、手順の明確さ、成功感などを 5 件法で評価。
- 負荷（簡易 NASA-TLX）: 心理的・時間的負荷を簡易的に記録。

4.6 ログ・記録

API リクエストとレスポンス、学習 `results.csv`/`args.yaml`、モデル切替履歴、クラッシュログを収集する。個人特定情報は収集しない。

4.7 倫理配慮とプライバシ

個人が特定される顔・氏名・住所などを含む画像を避け、社内・家庭内の共有ルールに従う。第三者が写り込む場合は事前同意を得る。

4.8 成功基準と終了条件

- 主要語彙（例: 食器/料理）で検出成立率が所定の閾値（例: 80
- 反復 2〜3 サイクル内に改善が頭打ちとなった場合は終了し、改善点を考察に残す。

4.9 想定リスクと緩和

- データ偏り: 撮影条件（明るさ/角度）を変化させる指示を追加。
- 語彙曖昧性: 類義語や重複概念は語彙定義を明確化。
- 計算資源: 学習は非同期・小規模から開始し、必要時のみ拡張。

4.10 結果の反映計画

成立率・反復時間・主観評価は表に集計し、改善サイクルの有効性を定性的に記述する。図は後に `Image_Goto/`へ出力・差し戻しを行う。

5

領域検出精度に関する実験

5.1 目的

本実験の目的は二つある。一つは、Semantic Segmentation (意味的領域分割) モデルでの残量領域の検出精度を確認することである。もう一つは、物体検出モデルでの食器の検出精度を確認することである。実験結果の各種評価指から、それぞれのモデルの比較を行い、その是非を考察する。

5.2 データセット

本実験で用いるデータセットは、食事中の写真を撮影し, ラベルを付けて作成した。このデータセットの画像サイズは 456(または 455) × 608、608 × 456(または 455) であり、モデルの入力画像は 224 × 224 にリサイズしており、全部で 241 枚で構成されている。その 241 枚の内、約 85%(206 枚) が学習のトレーニングに使用され、約 15%(35 枚) が学習のテストに使用される。領域分割タスクは、「ご飯」と「みそ汁」という二つのラベルを付けている (図 5.1)。物体検出タスクは、「食器」というラベルを付けている (図 5.2)。

図 5.1: 意味的領域分割タスク用のラベル　　図 5.2: 物体検出タスク用のラベル

5.3 実験条件

本実験では、以下の条件で学習、推論を行う (表 5.1)。各損失関数・評価関数の説明は節 5.4 で示している。

表 5.1: 学習・推論条件および実験環境

	項目	領域分割	物体検出
学習条件	学習エポック	100	
	バッチサイズ	4	
	学習率	1×10^{-3}	3×10^{-3}
	Optimaizer	Adam	
	Scheduler	Cosine annealing	
	損失関数	Tversky Loss	YOLOX Loss
推論条件	評価関数	mIoU	mIoU,mAP
実験環境	OS	Ubuntu 20.04 LTS	
	CPU	Dual Intel(R) Xeon(R) Gold 6342 CPU	
	GPU	Single NVIDIA RTX A6000 GPU	

5.4 評価指標

図 5.3 は、画像の分類対象に対して TP・FP・FN・TN の四領域に分割したものである。TP(True Positive) は、正解と予測して、実際に正解である領域。FP(False Positive) は、正解と予測して、実際は不正解である領域。FN(False Negative) は、不正解と予測して、実際は正解である領域。TN(True Positive) は、不正解と予測して、実際に不正解である領域。

図 5.3: 領域の分割の概要

5.4.1 Tversky loss

Tversky loss は、式 (5.1) で定義されている。FN と FP の割合 (α,β) を調整することで、重視したい要素に対して推測精度を向上させる狙いがある。本稿では、α=0.7, β=0.3 としている。

$$\begin{aligned} Tversky\_index(pred, gt) &= \frac{TP}{TP + \alpha FN + \beta FP} \\ Tversky\_Loss(pred, gt) &= 1 - Tversky\_index(pred, gt) \end{aligned} \tag{5.1}$$

5.4.2 IoU

Intersection over Union (IoU) は、式 (5.2) で定義されている。IoU は領域分割モデルや物体検出モデルの精度評価の時によく使われる。mIoU は、各クラスの IoU の平均を取っている。

$$\begin{aligned} IoU(pred, gt) &= \frac{TP}{TP + FN + FP} \\ IoU\_Loss(pred, gt) &= 1 - IoU(pred, gt) \end{aligned} \tag{5.2}$$

5.4.3 mAP

Average Precision (AP) を求める式は式 (5.3) で定義されている。Precision は、正解と予測したものの内、実際に正解だった割合である。Recall は、実際の正解の内、正解と予測した割合である。図 5.4 は、$p(r)$ をグラフにしたものの例である。図中の青色の線が Precision と Recall からプロットしたものであり、橙色はの線が積分を行うために変換したものである。変換先は、各 Recall より右にあるデータの内 Precision が最大な値となっている。そして、AP は、橙色の p(r) の離散積分で求める。例での AP の計算は式 (5.4) で示されている。mAP は、各クラスごとの AP の平均を取っている。AP_{num} となっている場合は、その数値の IoU を閾値として、それ以上での AP を算出している。

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ AP &= \int_0^1 p(r)dr \end{aligned} \tag{5.3}$$

図 5.4: P(r)

$$
\begin{aligned}
AP &= \frac{1}{11}(p(0) + p(0.1) + \cdots + p(1.0)) \\
&= \frac{1}{11}(1 \times 5 + 0.6 \times 4 + 0.55 \times 2) = 0.773
\end{aligned}
\tag{5.4}
$$

5.4.4 YOLOX loss

YOLOX Loss は式 (5.6) で定義されている。Box Loss は、正解ボックスと予測ボックスから IoU Loss を算出する。Objectness Loss は、物体の有無を予測したグリッドと実際のグリッドの物体の有無から BCE With Logits Loss を算出する。BCE With Logits Loss は、式 (5.5) で定義されている。式中の σ はシグモイド関数を示している。Classfication Loss は、それぞれのグリッドの予測クラスと正解クラスから BCE With Logits Loss から算出する。

$$
\begin{aligned}
\sigma(x) &= \frac{1}{1 + e^{-x}} \\
BCE\,With\,Logits\,Loss(pred, gt) &= gt \cdot \log \sigma(pred) + (1 - gt) \cdot \log(1 - \sigma(pred))
\end{aligned}
\tag{5.5}
$$

$$
\begin{aligned}
Box\,Loss &= \sum_i IoU(Box_{gt_i}, Box_{pred_i}) \\
Objectness\,Loss &= \sum_i BCE\,With\,Logits\,Loss(Obj_{pred_i}, Obj_{gt_i}) \\
Classfication\,Loss &= \sum_{i \in pred} BCE\,With\,Logits\,Loss(Cls_{pred_i}, Cls_{gt_i}) \\
YOLOX\,Loss &= 5.0 \times Box\,Loss + Objectness\,Loss + Classfication\,Loss
\end{aligned}
\tag{5.6}
$$

5.5 実験結果

表 5.2 はエンコーダ (バックボーン) の重みを固定したユニオンモデルでの実験結果の抜粋であり、図 5.5、5.6 はそれぞれ領域分割に対するバブルグラフと物体検出に対するバブルグラフである。表 5.3 は学習時のみエンコーダ (バックボーン) の重みを更新したユニオンモデル 2 の実験結果の抜粋であり、図 5.7、5.8 はそれぞれ領域分割に対するバブルグラフと物体検出に対するバブルグラフである。バブルグラフの横軸は Latency、縦軸はそれぞれ領域分割の mIoU・物体検出の mIoU、バブルの大きさはパラメータ量、バブルの色はバックボーン、バブルに紐づいている文字列は領域分割モデルを示している。mIoU は、モデルの精度であり、その数字が大きいほど (図中ではグラフの上になるほど) 精度が高いモデルといえる。FLoating-point Operations (FLOPs) は、浮動小数点演算の総量を示しており、その数字が小さいほど処理速度が速いモデルといえる。パラメータ量は、モデルの大きさであり、その数字が小さいほど (図中ではバブルが小さいほど) 小さいモデルといえる。Latency は、一枚の画像がモデルに入力されてから出力されるまでの時間の平均であり、その数字が小さいほど (図中ではグラフの右になるほど) 推論速度が速いモデルといえる。

図 5.5: 領域分割のバブルグラフ (ユニオンモデル)

図 5.6: 物体検出のバブルグラフ (ユニオンモデル)

Backbone	Method		Seg	Det		FLOPs	Params	Latency
	Seg	Det	mIoU[%]	mAP[%]	mIoU[%]	[G]	[M]	[ms]
MB_v3_s	U-Net	YOLOX	88.11	100.00	93.80	**0.84**	**3.67**	8.38
	DL3+		89.44			4.88	8.97	8.27
	DL3		75.97			0.85	8.27	8.23
EF_b0	U-Net	YOLOX	90.12	100.00	95.02	2.40	11.06	10.10
	DL3+		**91.13**			6.58	21.14	10.03
	DL3		77.01			2.55	20.44	9.99
R_18	U-Net	YOLOX	90.02	100.00	94.40	4.36	13.70	8.05
	DL3+		88.94			8.25	18.24	6.68
	DL3		72.85			4.21	17.54	**6.58**
R_152	U-Net	YOLOX	89.00	100.00	**95.83**	16.30	69.51	16.06
	DL3+		90.53			20.08	83.51	15.88
	DL3		74.87			16.02	82.79	15.43

表 5.2: 学習結果の抜粋 (ユニオンモデル)

Backbone	Method		Seg	Det		FLOPs	Params	Latency
	Seg	Det	mIoU[%]	mAP[%]	mIoU[%]	[G]	[M]	[ms]
MB_v3_s	U-Net	YOLOX	90	99.57	91.05	**0.84**	**3.67**	8.38
	DL3+		90.5	99.3	93.15	4.88	8.97	8.27
	DL3		83.35	100	92.67	0.85	8.27	8.23
EF_b0	U-Net	YOLOX	90.82	99.3	93.82	2.40	11.06	10.10
	DL3+		91.2	100	94.45	6.58	21.14	10.03
	DL3		85.05	99.29	94.71	2.55	20.44	9.99
EF_b4	U-Net	YOLOX	91.03	100	95.04	7.18	39.41	15.05
	DL3+		**91.33**	100	94.84	11.46	52.95	14.99
	DL3		84.1	100	**95.67**	7.43	52.25	14.90
R_18	U-Net	YOLOX	90.42	100	94.5	4.36	13.70	8.05
	DL3+		91.08	100	95.52	8.25	18.24	6.68
	DL3		84.13	100	95.57	4.21	17.54	**6.58**

表 5.3: 学習結果の抜粋 (ユニオンモデル 2)

5.6 考察

表・バブルグラフから読み取れる点として、バックボーンが同じ場合、Latency については大きな差は無く、FLOPs については Deeplabv3+が大きく U-Net と Deeplabv3 には大きな差は無く、パラメータ量については U-Net が小さく Deeplabv3 と DeepLabv3 ＋には大きな差は無い。領域分割の学習時のみバックボーンを学習する場合とバックボーンの重みを更新しない場合の領域分割精度を比較すると、U-Net と DeepLabv3 ＋では精度低下は小さいが、Deeplabv3 では精度低下が大きくなっている。これは、バックボーンの重みを更新しないことによる重み更新可能な層が少ない点が原因だと考えられる。また、バックボーンの重みを更新しない場合、ラベル数の少ない本稿のタスクでは、Deeplabv3+と U-Net はバックボーンを変更しても領域分割精度・物体検出精度は大きな差は無く、FLOPs・パラメータ量・Lantency の実行環境での使用可能な計算資源を加味して選択するのが望ましい。本稿のタスクでは、領域分割モデルの学習の時のみバックボーンの重みを更新しても物体検出モデルの精度に大きな影響を与えていないが、物体検出タスクが難しくなると領域分割に向けて学習されたバックボーンを使用すると精度低下は免れないと推測できる。そのため、より難しいタスクを行う場合は、バックボーンの重みを更新しないユニオンモデルの方が望まし

図 5.7: 領域分割のバブルグラフ (ユニオンモデル 2)

図 5.8: 物体検出のバブルグラフ (ユニオンモデル 2)

いと考えられる。以上のことから、計算資源が制限された環境でユニオンモデルを構築する場合、バックボーンとして MobiletNet v3 small・領域分割モデルとして U-Net・物体検出モデルとして YOLOX の tiny を選択したモデルが精度が高く、必要計算資源が小さいため適しているといえる。しかし、ただ必要計算資源の小さいモデルを選択するのが最適とは言えなく、使いたい環境での使用可能な計算資源に対して適しているモデルを選択するのが良いと考えられる。一方、計算資源が制限されていない環境でユニオンモデルを構築する場合、バックボーンとして Efficientnet b0・領域分割モデルとして Deeplabv3+・物体検出モデルとして YOLOX の tiny を選択したモデルが精度が高く適していると言える。

本稿では、判別するクラスを絞っているため、判別するクラスを増加したモデルの構築と、更なる精度向上のためにより良いモデルの探索やより良い学習率・エポック数などの学習条件の探索は今後の課題である。

6

残量測定に関する実験

6.1 目的

本実験の目的はの残量推定精度を確認することである。実験結果からそれぞれの推定手法の比較を行い、その是非を考察する。

6.2 実験条件

6.2.1 推定用データセット

本実験で用いるデータセットは、食事中の写真を撮影し, それぞれの数値を測定した (図 6.1)。数値は食事前の画像を 100%としている。このデータセットの画像サイズは 1477 × 1109 であり、全部で 34 枚で構成されている。

図 6.1: 画像とその数値 [%]

6.2.2 評価指標

評価指標は実測値 [%] と推定値 [%] の平均二乗誤差 (MSE) の平方根 (RMSE) を用いている (式 (6.1))。

$$\begin{aligned} MSE(pred, gt) &= \frac{1}{n}\sum_{i=1}^{n}(pred_i - gt_i)^2 \\ RMSE &= \sqrt{MSE} \end{aligned} \tag{6.1}$$

6.3 実験結果

表 6.1 はエンコーダ (バックボーン) の重みを固定したユニオンモデルでの測定結果の抜粋であり、図 6.2,6.3,6.4,6.5 は前者は球冠ベース・後者は関数ベースでの測定結果の RMSE のバブルグラフである。バブルグラフの横軸は領域分割の mIoU、縦軸は物体検出の mIoU、バブルの大きさは RMSE の大きさ、バブルの色はバックボーン、バブルに紐づいている文字列は領域分割モデルを示している。領域分割の mIoU は、モデルの領域分割精度であり、その数字が大きいほど (図中ではグラフの左になるほど) 精度が高いモデルといえる。物体検出の mIoU は、モデルの物体検出精度であり、その数字が大きいほど (図中ではグラフの上になるほど) 精度が高いモデルといえる。RMSE は、推定の精度を示し、その数字が小さいほど (図中ではバブルが小さいほど) 推定精度のよいものといえる。

Backbone	Method		Seg	Det	球冠ベース		関数ベース	
	Seg	Det	mIoU[%]	mIoU[%]	ご飯 [%]	みそ汁 [%]	ご飯 [%]	みそ汁 [%]
MB_v3_s	Unet	YOLOX	88.11	93.80	9.10	29.62	9.90	25.30
EF_b0	DLv3+	YOLOX	**91.13**	95.02	12.21	31.78	12.64	23.56
EF_b6	Unet	YOLOX	89.64	**95.79**	14.03	**18.70**	14.57	**12.78**
R_18	Unet	YOLOX	90.02	94.40	**7.56**	20.82	**8.44**	17.01

表 6.1: 測定結果の抜粋 (ユニオンモデル)

図 6.2: 球冠ベース (ご飯) のバブルグラフ　図 6.3: 球冠ベース (みそ汁) のバブルグラフ

6.4　考察

表・バブルグラフから読み取れる点として、どちらの手法 (球冠ベース・関数ベース) でも、領域分割精度・物体検出の精度が悪いと、その結果から推定するため RMSE が大きくなる。また、どちらの手法でもご飯の方がみそ汁よりも RMSE が小さくなっている。これはみそ汁の淵が透明であり、その結果みそ汁に対する領域分割精度が低下したためと考えられる。加えて、モデルの精度が低い場合、空になった食器に残る液体をみそ汁と検出していることが RMSE が大きくなった原因として考えられる。ご飯に対する精度はご飯の盛り上がり具合に依存しており、どちらの手法も盛り上がりを考慮していないため、盛り上がりが大きいほど RMSE が大きくなると考えられる。本実験の球冠ベースと関数ベースを比較すると、ご飯に対しては球冠ベース・みそ汁に対しては関数ベースの手法の方が推定精度が高い。この結果から、対象の食器に対して測定手法を変更することが良いと考えられる。

今実験では、mIoU の精度が高いが出力画像の精度が高くない場合があった。これは、学習の画像と評価用の画像での画像サイズの違いや画像の輝度の違いが理由として考えられる。また、学習データセットの枚数が 241 枚と少なく、過学習となり汎化性能が落ちた可能性も考えられる。

本稿では、先述したとおり、ご飯の盛り上がりやへこみを考慮しておらず、食器の近似にも限界があるため、領域分割モデルの特徴マップから残量推定を行う仕組みの構築による推定精度向上が今後の課題である。

図 6.4: 関数ベース (ご飯) のバブルグラフ　図 6.5: 関数ベース (みそ汁) のバブルグラフ

7

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。意味的領域分割モデルと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、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. **Haruhhiro Takhashi**, “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	Det		FLOPs	Params	Latency
	Seg	Det	mIoU[%]	mAP[%]	mIoU[%]	[G]	[M]	[ms]
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	Det		FLOPs	Params	Latency
	Seg	Det	mIoU[%]	mAP[%]	mIoU[%]	[G]	[M]	[ms]
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	Det	球冠ベース		関数ベース	
	Seg	Det	mIoU[%]	mIoU[%]	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	Det	球冠ベース		関数ベース	
	Seg	Det	mIoU[%]	mIoU[%]	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)