

# 深層学習による画像認識を用いた 低波浪時の離岸流発生場所把握のための研究

長岡技術科学大学大学院工学研究科 環境社会基盤工学分野 修士2年 田邊 修斗

## 1. はじめに

毎年、海辺での溺水事故が発生しているがその原因の一つとして離岸流がある。日本ライフセービング協会によると、海水浴場における溺水事故の約50%が離岸流に起因している<sup>1)</sup>。離岸流は、海岸付近での波浪と地形の作用で発生する沖方向への強い流れであるが、一般には発生場所を把握するのは困難であり、海岸での監視者の減少とあわせて事故発生数減少に至らない原因となっている。これに対応するために、Dumitriu ら<sup>2)</sup>は『A Novel Benchmark and YOLOv8 Baseline Results』で離岸流のインスタンスセグメンテーション、石川ら<sup>3)</sup>は『特徴の異なる離岸流を検知可能なAIモデルの構築』でオープンエリアの離岸流と突堤付近の離岸流を検知可能なモデルの構築に取り組むなど、近年ではAIによる離岸流の発生状況の判別手法が開発されており実用化されつつあるが、他既往成果例では、砕波現象が小さい低波浪時での判別は困難である。しかし本研究グループの既往成果<sup>4)</sup>より、低波浪時の海水浴中の事故は全体の半数で発生していることから、低波浪時の判別を可能にさせる必要がある。そこで本研究では、本グループが離岸流研究から得た知見を活用し、砕波現象が小さい海象時における離岸流発生場所における表面波の変形パターン（図-1）を画像認識し、深層学習モデルを構築し、離岸流の発生を検知することを試みた。

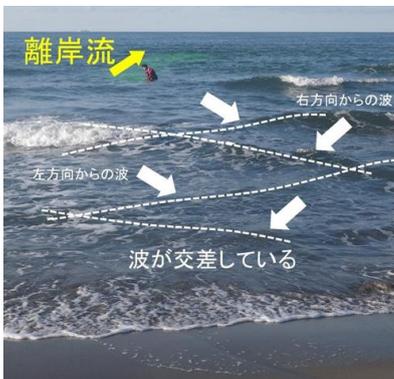


図-1 離岸流発生時の表面波の変形パターン

## 2. 手法

映像から物体を検出する手法は様々提案されている。本研究では、深層学習を利用した一般物体検出アルゴリズムの中でも高速処理が可能な、YOLO<sup>5)</sup>の後続システムである YOLOv8<sup>6)</sup>を利用した。

YOLO の物体認識の手法は、予め画像全体を正方形のグリッドに分割しておき、それぞれのグリッドに対象となる物体が含まれているかどうかを判定するものである。

手順は以下の通りである。

- ①：入力された正方形の画像をさらにグリッドセル(grid cell)と呼ばれる小さな  $S \times S$  個の正方形に分割する。
- ②-1：各正方形に任意の個数のバウンディングボックスを配置する。
- ②-2：各正方形が、どのクラスに所属している可能性が高いかを判断する。
- ③：バウンディングボックスから適切なものを選ぶために、クラスごとに一番信頼度の高い枠を選び、その枠と他の枠の重なり具合を調べて一定以上の割合で重なっている枠を消し、物体を検出する。

## 3. モデルの構築

### (1) 使用データ

教師データに用いたのは、藤塚浜海岸において 2015 年に撮影された離岸流発生映像と、Dumitriu ら<sup>2)</sup>により公開されているデータセットである。Dumitriu らのデータセットを教師データに用いたのは、Dumitriu らデータセットが、低波浪時の離岸流検知に適用できるかについて検討するためである。また、藤塚浜海岸における撮影時の波高、周期、波向はそれぞれ、0.42m, 4.1s, NW であった。藤塚浜海岸の映像に対してはフレームに分解し、離岸流発生場所に顕著に見られる表面波の変形パターンをアノテーションした。作成したデータセットは以下の5つである。

表-1 データセット

dataset1	: 藤塚浜海岸_本研究学習データ150枚
dataset2	: 藤塚浜海岸_本研究学習データ500枚
dataset3	: 藤塚浜海岸_本研究学習データ630枚
dataset4	: Dumitriu <sup>2)</sup> の公開データセット2466枚
dataset5	: Dataset3とDataset4,組み合わせ3096枚

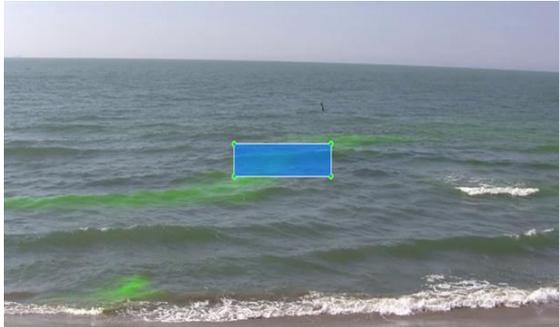


図-2 離岸流箇所のアノテーション例

## (2) 使用ツール

本研究では、Google が提供しているクラウドベースの Python 実行環境である Google Colaboratory を使用した。Google Colaboratory はブラウザ上で動作し、環境構築の必要がなく、また GPU (Tesla T4) へのアクセスが可能である。そのため、計算リソースの制約を受けずに深層学習の学習や検証を行うことができる。

## (3) 学習条件

モデル: YOLOv8, エポック数: 300, 学習率 0.01, 訓練データと検証データの分割 7:3, バッチサイズ: 8, 画像サイズ: 640×640 ピクセル, 最適化手法: auto (学習率やその他パラメータを自動的に調整する手法), ライブラリ: pytorch, メモリ容量: 14.75GB

## 4. 結果

### (1) 検証データセットに対する評価

dataset1からdataset5までのそれぞれをモデルに学習させた後、訓練データセットとは異なる検証データセットを作成しモデルのパフォーマンスを確認した。テストデータセットは2つ作成した。教師データ作成に用いた画像と同じアングルの画像テストデータセットと、別の角度の画像データからなる2つのテストデータセットである。

データセット作成後、Precision, Recall, mAP50, mAP50-95を算出し、dataset4の低波浪時における適用可能性を確認するとともに、dataset3とdataset5の性能の比較を行った。なお、

本研究は、水難事故防止の観点から、離岸流発生箇所を把握することが目的である。そのため、離岸流の検出漏れを最小限に抑えることが重要であり、Recallを重視することとした。

表-2 テストデータセット

テストデータセット①	: 5401枚 (教師データと同じアングル)
テストデータセット②	: 1798枚 (教師データと別のアングル)

図-3 に、同じアングルから撮影されたデータセット 5401 枚での検証の結果を示している。dataset4 では Precision, Recall, mAP50, mAP50-95 のすべての指標が 0 となり、低波浪時の離岸流を適切に把握できていないことが示された。また dataset1 では各指標が 0 であったが、dataset2, dataset3 では、藤塚浜海岸での教師データ画像の増加に伴って Recall が向上した。dataset5 においては、藤塚浜の離岸流データを最も多くした dataset3 と比較して、Precision は低下しているものの、Recall が向上した。Precision と Recall は、一般的にトレードオフの関係にある。Precision が高い場合、モデルが正確に離岸流を検出できるが、検出漏れが発生する可能性が高くなり、Recall が低下する。逆に、Recall が高い場合、モデルが多く離岸流を検出できるが、誤検出が増えて Precision が低下する。先にも述べたように、本研究では Recall を重視している。よって Dumitriu らのデータセットと本研究学習データを複合的に用いた dataset5 モデルの汎用性が向上することが確認できた。

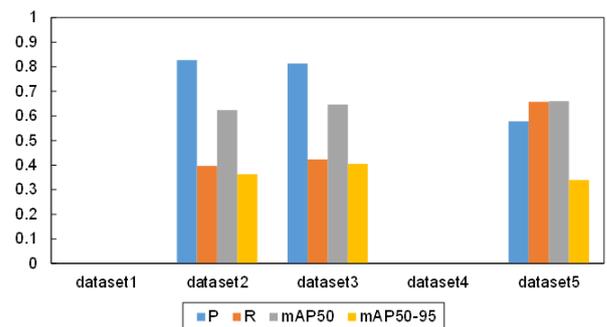


図-3 同じアングルでの検証結果

テストデータセット②での検証結果について図-4 に示している。図-4 より、dataset1 から dataset4 では適切に離岸流の発生箇所を把握で

まず、各指標が0になっていることが確認できる。一方で、dataset5では各指標の向上が見られた。よって、dataset5においては、他のデータセットと比較して汎用性が向上し、別のアングルでの離岸流についても判別できる可能性が示された。

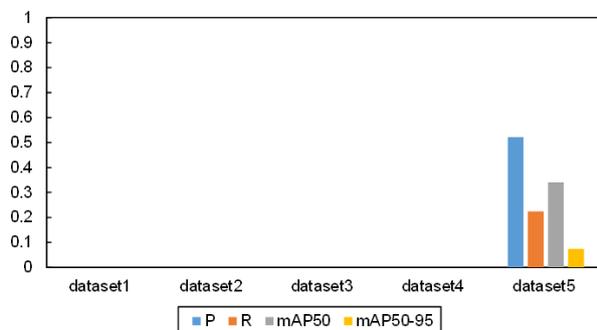


図-4 別のアングルでの検証結果

### (2) 動画に対する予測結果（藤塚浜海岸）

訓練済みのモデルを使用して新しいデータの出力を予測した。ここでは、dataset3とdataset5の検出率の比較を行った。(1)では、検証用に画像にアノテーション情報を加えた検証用データセットを構築し評価を行ったが、動画に対して検証を行う際、離岸流の発生するすべてのフレームに対して人の手でアノテーションを施し、評価を行うのは現実的ではない。よって、動画に対しての予測検証では、動画全体で離岸流が発生しているものとし（着色剤発生は確認済み）、検出フレーム数で評価を行った。なお、00009.mp4, 00012.mp4, 00013.mp4は別のアングル、00015.mp4は同じアングルの映像である。

結果、dataset5ではdataset3と比較して検出率が向上していることが確認できた。一方、検出率は向上しているものの、動画によっては約18%しか検出していないことが確認できた。これは、本研究で構築したモデルは離岸流発生時の表面波の交差パターンが顕著な画像をアノテーションし、教師データとしているが、離岸流が発生しているすべてのフレームにおいて離岸流の顕著なパターンが出現しているとは限らない（1波浪のうち瞬間的）ためである。そこで、動画を解析している間、4秒（約1周期）ごとに離岸流が検出された場合に、それを1回としてカウントし、動画全体での4秒ごとの検出率を算出した。その結果を図-6に示している。dataset3では同じアングルの映像に対しては検出率が高い一方で、別のアングルの映像では検

出率が低くなった。dataset5では、同じアングルだけでなく、別のアングルの映像に対しても検出率が向上した。

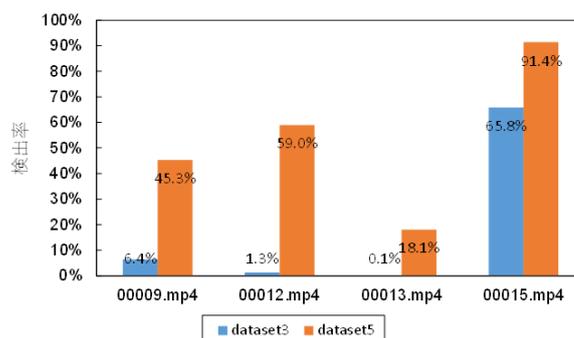


図-5 総フレームに対する検出率（藤塚浜海岸）

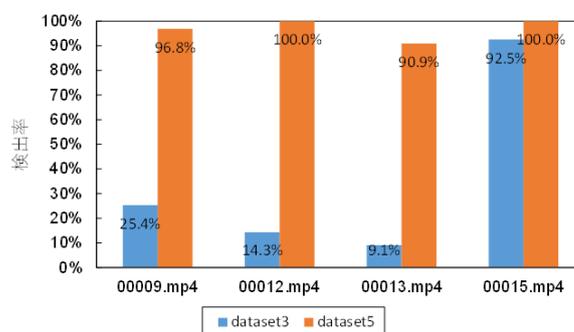


図-6 4秒ごとの検出率（藤塚浜海岸）

### (3) 動画に対する予測結果（内灘海岸）

ほかの海岸での適用可能性について確認した。これには石川県内灘海岸の離岸流発生映像を用いた。内灘海岸での検証についても、藤塚浜海岸での予測と同様に、総フレームに対する検出率、4秒ごとの検出率を算出した。

なお、内灘海岸での離岸流撮影期間、波浪条件は以下の通りである。

表-3 内灘海岸における撮影期間と波浪条件

内灘①, ②	2018年6月28日, 1.05m, 5.5s, W
内灘③, ④	2019年7月 8日, 0.51m, 4.0s, N

図-7, 図-8に結果を示しているが、内灘海岸での予測でも、藤塚浜海岸に対する予測と同様にdataset5で検出率が向上していることが確認された。

また、実際に予測動画を確認し、離岸流の発生場所の確からしさを確認した。図-9に検出の一例を示しているが、この画像より、離岸流の発生場所を正しく検出していることが確認できる。よって、本研究で構築したモデルは、異な

る海岸，異なる波高時の離岸流を検出することができることが確認できた。

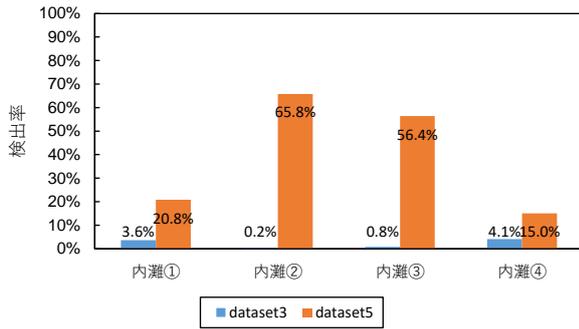


図-7 総フレームに対する検出率（内灘海岸）

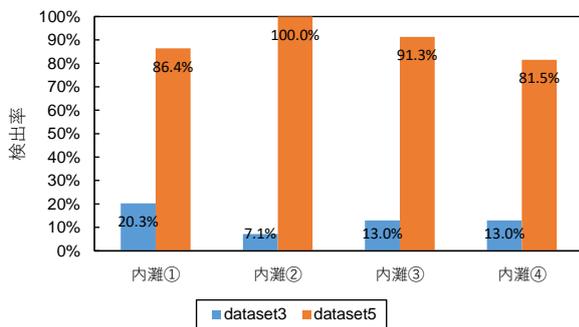


図-8 4秒ごとの検出率（内灘海岸）



図-9 内灘海岸離岸流検出例

## 5. まとめ

### 1) 検証結果

検証データセットに対するテストにより，Dumitriu らのデータセットでは低波浪時の離岸流を検知することはできないことが確認できた。一方で，藤塚浜学習データと Dumitriu らのデータを複合的に用いることで再現率が最も高くなった。

### 2) 動画に対する予測（藤塚浜海岸）

藤塚浜学習データ 630 枚を用いたデータセット（dataset3）では同じアングルの離岸流映像

に対して，4 秒ごとの検出率では 92.5% で離岸流を認識した。また，Dumitriu らのデータセットと藤塚浜海岸の離岸流映像から構築したデータセットを複合的に用いたデータセット（dataset5）では，同じアングル映像に対しては 100% 認識し，カメラのアングルが変化した場合にも，低波浪時の離岸流発生箇所を 90% 以上検出した。

### 3) 動画に対する予測（内灘）

内灘での検証においても dataset5 で検出率が向上していることが確認できた。これにより，別の海岸，異なる波高においても離岸流を検出することができることを確認できた。

## 6. 今後の展望

- ・海岸全体と離岸流発生場の画面内の適度な比率を考える必要があると考えられる。
- ・内灘海岸の映像は短時間であったため，異なる波高時の判別をさらに確認する必要があると考えられる。

## 7. 参考文献

- 1) 日本ライフセービング協会，<https://jla-lifesaving.or.jp/watersafety/ripcurrent/>，（最終閲覧日:2024年1月22日）
- 2) Andrei Dumitriu, Florin Tatui, Florin Miron, Radu Tudor Ionescu, Radu Timofte.: Rip Current Segmentation: A Novel Benchmark and YOLOv8 Baseline Results. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023.
- 3) 石川 仁憲，島田 良，Mika Viljam KONTTO，小峯 力：特徴の異なる離岸流を検知可能な AI モデルの構築，土木学会，土木学会論文集 B2，Vol.78，No.2.
- 4) 犬飼 直之，江尻 義史，大竹 剛史，山本 浩，細山田 得三：新潟東港周辺における突堤で囲まれたカस्प地形海岸での離岸流の生成機構について，土木学会，土木学会論文集 B2，Vol.71，No.2.
- 5) Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection, 2016.
- 6) Ultralytics: YOLOv8 in PyTorch, <https://docs.ultralytics.com/>, (最終閲覧日:2024年1月30日)