

ドローン画像と物体検出 AI を用いた、笹ヶ瀬川流域におけるごみ分布推定

株式会社 Solafune

2025 年 2 月 21 日

要旨

株式会社 Solafune は、ドローンで撮影した河川の画像を活用し、ごみを検出する物体認識モデルを開発した。従来の目視による確認では膨大な時間と労力がかっていたが、本モデルを導入することで検出精度と作業効率の両面で大きな改善が見込める。本プロジェクトの実証においても、対象となる河川のごみを従来よりも迅速に検出できることを確認した。

さらに、検出したごみの分布密度を地図上に可視化するシステムを構築し、画像から可視化への作業工程を半自動化している。この取り組みによって、大量の画像データを効率よく処理し、どの地点にどの程度のごみが存在するのかを一目で把握できるようになった。これにより、清掃作業の優先度や実施スケジュールを柔軟に検討できるだけでなく、今後の河川環境保全計画にも重要な指針を提供できると考えている。

1. はじめに

世界各地から大量のごみが海洋に流出している現状は、現代の重大な環境問題の一つである。海に流れ出たごみの回収はきわめて困難であり、そもそもの流出量を削減することが喫緊の課題となっている。そのためには、海に到達する前の河川域におけるごみの分布状況を調査し、河川内の段階で有効な対策を講じると同時に、河川近隣住民への啓発活動を強化する必要がある。そのためには、実際のごみの分布を知り適切な対策を打つことが重要である。しかし、河川のごみ分布を定量的に調べるには多大なコストと労力がかかるという問題があった。そこで、株式会社 Solafune（以下、Solafune）は、物体検出 AI を活用してごみの分布推定を効率化する取り組みを行った。

2. 手法

本プロジェクトでは、効率的に河川のごみ分布を調べるためにドローンによる笹ヶ瀬川流域の空撮データを使用し、そのデータをごみの検出に特化した物体検出 AI で解析した。

2.1 ドローンデータ

撮影には Phantom 4 Advanced というドローンを使用し、静止画と動画の両方を取得した。具体的には、左岸を画像で、右岸を動画で撮影し、それぞれの検出結果を比較することで解析時の特性を理解し、今後の別地域での応用に役立てることを目的とした。撮影は笹ヶ瀬川全体を 17 のブロックに分割したエリアに対して 2024 年 11 月 11-13 日の三日間で実施し、最終的に 2,590 枚の写真と約 20 本の動画データを取得した（図 1、表 1）。本撮影によって画像と動画の両データを取得することができ、それぞれでの検出精度や運用上の利点を比較検討し、より効率的な調査手法を確立するための知見を得ることを目指している。

2.2 物体検出 AI による調査（画像）

物体検出モデルとして、軽量で物体検出の精度が高い YOLO¹を採用した。本プロジェクトで取り扱う画像は高解像度でありながら、実際に写り込んでいるごみが非常に小さいという特徴を持つ。

¹ <https://github.com/ultralytics/ultralytics>

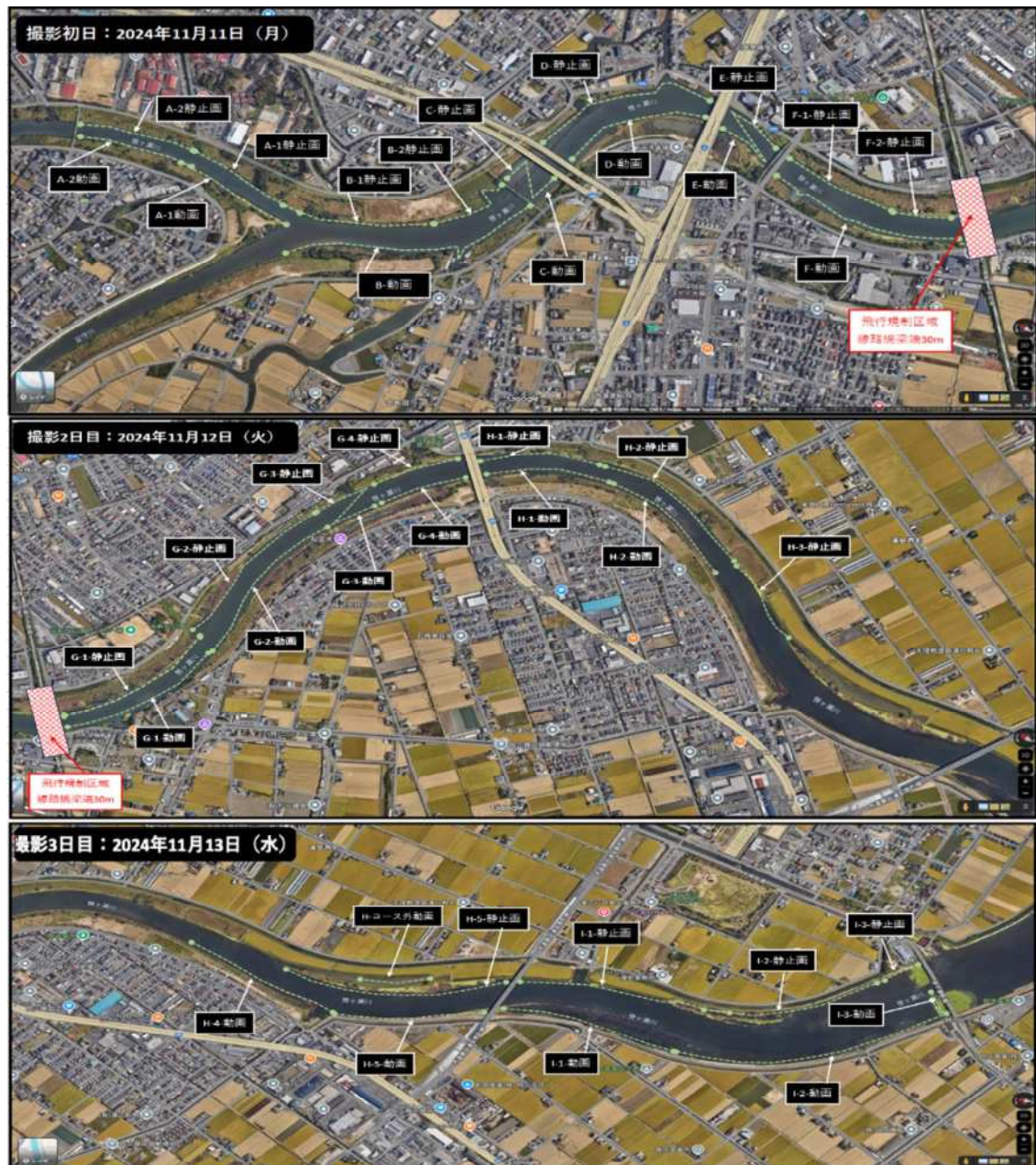


図1 ドローン撮影エリアの名称

汎用性の高いシステムの構築のためには、高解像度画像を扱う能力を維持しつつ、Google Colab 等の GPU でも実行時間を過度に増やさないモデルとして、YOLO11m を選択した。YOLO11 は YOLO の中で解析当時最新のモデルであり、その中でモデルの大きさごとに比較した結果、YOLO11m が解析時間と解像度とのバランスが適していた。

既存の学習済みモデルをそのまま利用した場合、多くのごみが見落とされるため、追加の学習を行うことでごみ検出に特化した独自のモデルを構築した。学習に使用したデータは、ごみ画像を中

心とした TACO dataset²を利用した。さらに、本プロジェクトで用いるドローン空撮画像では、ごみがフレーム内で極めて小さく写っているケースが多いという特性を持っているため、実際に撮影したドローン画像の一部（20 枚）にアノテーションを施し、それらを学習に組み込んだ。物体の分類方法は TACO dataset に準じ、以下のようなクラスで分類した。

表1 ドローンの撮影ログ

笹ヶ瀬川両岸ゴミ撮影（無人航空機写真撮影）			
フライトログ（飛行記録） Monday, November 11, 2024 15データ			
1日目	No	飛行コース	データ名
	1	A-1動面	2024-11-11_09-10-50_v2
	2	飛行のみ	2024-11-11_09-36-32_v2
	3	A-1静止面	2024-11-11_09-38-54_v2
	4	A-2動面	2024-11-11_10-18-44_v2
	5	A-2静止面	2024-11-11_10-34-38_v2
	6	B-1静止面	2024-11-11_11-38-59_v2
	7	B-2静止面	2024-11-11_11-58-19_v2
	8	B-動面	2024-11-11_13-08-07_v2
	9	C-動面・静止面	2024-11-11_13-30-35_v2
	10	D-動面	2024-11-11_14-00-03_v2
	11	D-静止面	2024-11-11_14-12-13_v2
	12	E-動面・静止面	2024-11-11_14-42-55_v2
	13	F-動面	2024-11-11_16-04-53_v2
	14	F-1静止面	2024-11-11_16-25-32_v2
	15	F-2静止面	2024-11-11_16-47-37_v2
フライトログ（飛行記録） Tuesday, November 12, 2024 13データ			
2日目	No	飛行コース	データ名
	1	G-1静止面	2024-11-12_09-01-56_v2
	2	G-1動面	2024-11-12_09-26-44_v2
	3	G-2静止面	2024-11-12_09-59-42_v2
	4	G-2動面	2024-11-12_10-23-39_v2
	5	C-静止面・動面	2024-11-12_11-00-46_v2
	6	G-4静止面	2024-11-12_11-28-03_v2
	7	G-4動面	2024-11-12_11-46-42_v2
	8	H-1静止面	2024-11-12_13-15-52_v2
	9	H-1動面	2024-11-12_13-37-17_v2
	10	H-2静止面	2024-11-12_15-03-35_v2
	11	H-2動面	2024-11-12_15-27-37_v2
	12	H-3静止面	2024-11-12_16-02-39_v2
3日目	13	飛行のみ	2024-11-12_16-26-04_v2
	14		
	15		
フライトログ（飛行記録） ##### 11データ			
3日目	No	飛行コース	データ名
	1	H-4動面	2024-11-13_08-47-32_v2
	2	H-5静止面	2024-11-13_09-19-04_v2
	3	H-5動面	2024-11-13_09-43-25_v2
	4	I-1静止面	2024-11-13_10-15-38_v2
	5	I-1動面	2024-11-13_10-40-13_v2
	6	I-2静止面	2024-11-13_11-19-03_v2
	7	I-3静止面	2024-11-13_11-43-55_v2
	8	I-2動面	2024-11-13_11-52-56_v2
	9	I-3動面	2024-11-13_13-17-28_v2
	10	H-コース内 飛行のみ	2024-11-13_13-48-42_v2
	11	H-コース内 I-1動面	2024-11-13_13-56-26_v2
	12		
	13		
	14		
	15		

² <http://tacodataset.org/>

表2 学習に用いた Class ID と名称

Class ID	Item	Class ID	Item
0	Aluminium foil (アルミホイル)	31	Tissues (ティッシュ)
1	Battery (バッテリー)	32	Wrapping paper (包装紙)
2	Aluminium blister pack (アルミブリストパック)	33	Normal paper (普通紙)
3	Carded blister pack (カードブリストパック)	34	Paper bag (紙袋)
4	Other plastic bottle (その他のプラスチックボトル)	35	Plastified paper bag (プラスチック加工紙袋)
5	Clear plastic bottle (透明プラスチックボトル)	36	Plastic film (プラスチックフィルム)
6	Glass bottle (ガラス瓶)	37	Six pack rings (シックスパックリング)
7	Plastic bottle cap (プラスチック製ボトルキャップ)	38	Garbage bag (ゴミ袋)
8	Metal bottle cap (金属製ボトルキャップ)	39	Other plastic wrapper (その他のプラスチック包装)
9	Broken glass (破損ガラス)	40	Single-use carrier bag (使い捨て買い物袋)
10	Food Can (食品缶)	41	Polypropylene bag (ポリプロピレンバッグ)
11	Aerosol (エアロゾル)	42	Crisp packet (スナック菓子袋)
12	Drink can (飲料缶)	43	Spread tub (スプレッド容器)
13	Toilet tube (トイレチューブ)	44	Tupperware (タッパーウェア)
14	Other carton (その他のカートン)	45	Disposable food container (使い捨て食品容器)
15	Egg carton (エッグカートン)	46	Foam food container (発泡スチロール食品容器)
16	Drink carton (ドリンクカートン)	47	Other plastic container (その他のプラスチック容器)
17	Corrugated carton (段ボール)	48	Plastic gloves (プラスチック手袋)
18	Meal carton (ミールカートン)	49	Plastic utensils (プラスチック製食器)
19	Pizza box (ピザ箱)	50	Pop tab (ポップタブ)
20	Paper cup (紙コップ)	51	Rope & strings (ロープ・ひも)
21	Disposable plastic cup (使い捨てプラスチックカップ)	52	Scrap metal (金属くず)
22	Foam cup (発泡スチロールカップ)	53	Shoe (靴)
23	Glass cup (ガラス製カップ)	54	Squeezable tube (押し出しチューブ)
24	Other plastic cup (その他のプラスチックカップ)	55	Plastic straw (プラスチックストロー)
25	Food waste (食品廃棄物)	56	Paper straw (紙ストロー)
26	Glass jar (ガラスジャー)	57	Styrofoam piece (発泡スチロール片)
27	Plastic lid (プラスチック製蓋)	58	Unlabeled litter (ラベル無しごみ)
28	Metal lid (金属製蓋)	59	Cigarette (タバコの吸い殻)
29	Other plastic (その他のプラスチック)		
30	Magazine paper (雑誌紙)		

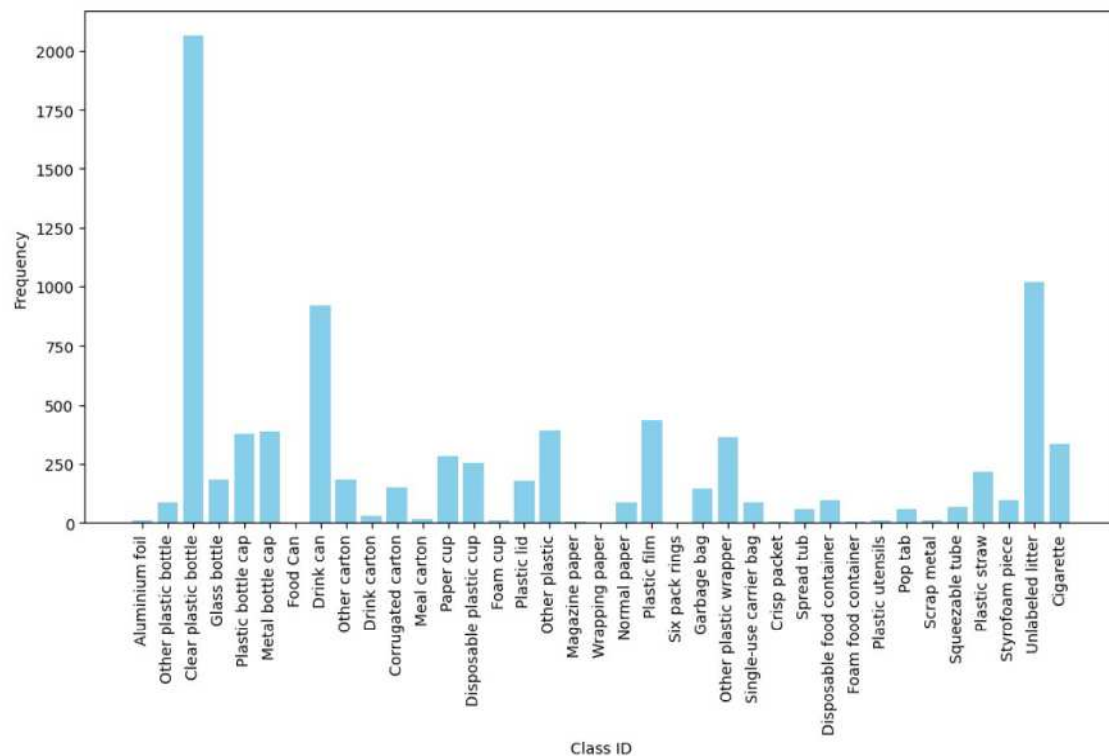


図2 クラスごとの検出頻度

本プロジェクトで行なった学習は、事前学習済みモデルyolo11m.ptをベースに、データセットを指定し、300 エポックにわたる学習を実施した。画像サイズは1280px、5 エポックごとに中間モデルを保存し、途中結果の検証や異常時の復元を可能にした。実際に利用したものと同様のプログラムは Appendix に Google Colab 形式で共有している。

TACO dataset (1,500 枚) とドローン画像 (20 枚) を合わせたデータを、それぞれ学習用と検証用に 4:1 の比率で分割し、合計 200 エポックの学習を実施した。最終的に、検証スコアが最も良いモデルを採用してドローンデータからごみの検出を行なった。本システムは、TACO データセットを用いて学習された物体検出モデルを使用し、画像内の対象物を検出する。解析対象の画像は、事前に指定したディレクトリ内のすべての JPEG ファイルであり、それらを読み込んで EXIF データを取得し、GPS 情報やカメラの向きを特定する。

検出プロセスでは、画像を一定サイズ (1280px) のウィンドウに分割し、各領域に対して物体検出を適用する。検出されたオブジェクトの位置は、ドローンの GPS 情報やカメラの向き、撮影高度を基に計算し、地表座標に変換する。また、重複検出を防ぐために IoU (Intersection over Union) を利用して類似領域を排除する。最終的に、検出されたオブジェクトの座標、クラス ID、信頼度を記録し、検出結果を画像として保存するとともに、CSV ファイルにログとして記録する。これにより、検出結果を後処理しやすくし、位置情報を活用した分析が可能となる。

各ドローン画像から抽出された検出ログファイルをもとに、撮影エリアにおけるごみの分布を正確に評価するための後処理を実施した。ログファイルには、ドローンの撮影位置、カメラ中心座標、検出されたごみの地上座標、クラス ID、信頼度、さらに画像の四隅の地上座標が記録されている。これらの検出結果から、同一画像内においては各検出を個別の物体として扱う一方で、異なる

表3 各クラスの検出数

Class ID	名称	検出数
0	Aluminium foil	10
4	Other plastic bottle	85
5	Clear plastic bottle	2065
6	Glass bottle	185
7	Plastic bottle cap	376
8	Metal bottle cap	384
10	Food Can	2
12	Drink can	918
14	Other carton	182
16	Drink carton	28
17	Corrugated carton	151
18	Meal carton	14
20	Paper cup	283
21	Disposable plastic cup	252
22	Foam cup	10
27	Plastic lid	177
29	Other plastic	392
30	Magazine paper	4
32	Wrapping paper	1
33	Normal paper	87
36	Plastic film	434
37	Six pack rings	2
38	Garbage bag	148
39	Other plastic wrapper	364
40	Single-use carrier bag	84
42	Crisp packet	5
43	Spread tub	56
45	Disposable food container	92
46	Foam food container	6
49	Plastic utensils	9
50	Pop tab	58
52	Scrap metal	12
54	Squeezable tube	65
55	Plastic straw	215
57	Styrofoam piece	93
58	Unlabeled litter	1018
59	Cigarette	335
ALL	総計	8602

画像で撮影された重複領域において、同一のクラス ID を持ちかつ一定の距離（2メートル以内）にある検出については、信頼度の低い側に重複フラグを付与することで、実際には同一物体であるとみなした。これにより、複数の画像で重複して検出された場合にも、ユニークな物体数を正確にカウントすることが可能となった。

さらに、各画像の四隅の座標を用いて、個々の撮影エリアをポリゴンとして定義し、シェイプリーの合併処理を適用することで、重複部分を除いた全体の撮影エリアを抽出した。ログ上の座標は緯度経度（EPSG:4326）の形式で記録されているため、正確な面積計算のために Web Mercator（EPSG:3857）へ投影し、合併ポリゴンの面積（平方メートル）を算出した。最終的に、重複を除いたユニークなごみの検出数を、撮影エリアの正味面積（km²換算）で割ることで、地域全体の平均ごみ密度（個/km²）を求めるとともに、得られた結果を基に現場のごみ分布状況の定量的評価を行った。

ログファイルの中から、重複を除いたごみを全て集計すると、撮影した領域の総面積は約 11 万平方メートル、撮影した河川の延長約 9.8km で、ごみの検出総数は 8602 個だった（表）。これは、約 13 平方メートルに 1 個ごみが検出されたことに相当する。また、その中でもペットボトルと缶の検出数が突出して多く、それぞれ 2065 個と 918 個だった（図 2、図 3）。

表4 撮影、検出のまとめ

画像枚数(枚)	2590
撮影総面積(km ²)	0.112
撮影総距離(m)	9.765
ゴミ検出総数(個)	16191
重複除外したゴミの総数(個)	8602

2.3 物体検出 AI による調査（動画）

本プロジェクトでは、静止画に加えて動画データからの物体検出も試みた。動画解析では、動画の各フレームを抽出し、画像と同様の物体検出プロセスを適用する予定であった。しかし、静止画とは異なり、動画にはフレームごとの座標情報が直接含まれていないため、フライトログとのマッチングが不可欠であった。

このマッチング作業には、動画撮影開始時刻とフライトログを同期させ、各フレームの検出結果に正確な位置情報を付与する必要があったが、このプロセスの自動化には予想以上の工数を要し、実装に至らなかった。また、手作業でのマッチングも試みたが、処理量が膨大であり、限られた期間内での完遂が困難であった。

加えて、技術的には静止画解析と大きな差異がないことから、より効率的なデータ処理を優先し、解析負荷の低い静止画を中心に分析を進める方針を採用した。結果として、静止画解析においては十分な精度で検出が行えたため、本プロジェクトでは動画データの解析に関しては一部のバウンディングボックス付き動画の出力に留める形となった。

3. ごみの分布の可視化

ごみの地図上の座標が記録されているログファイルから、Google Maps 上に密度分布を表すヒートマップを作成した（図 2）。ログファイルには、検出されたごみのカテゴリ（ペットボトル、空き

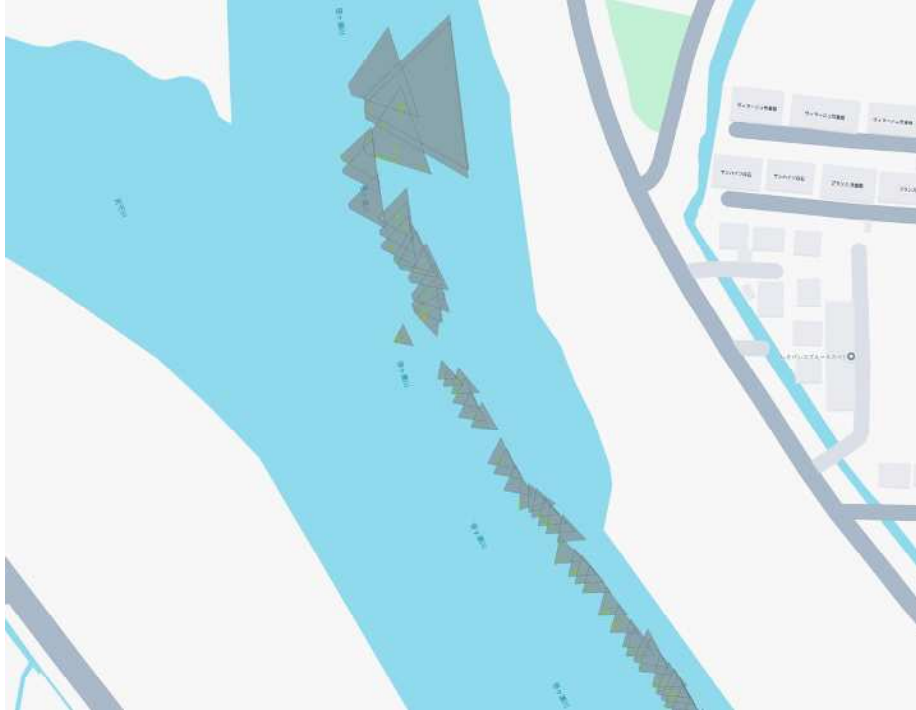


図3 Google Maps 上に描画されている図。台形が画像に写っているエリアを表す

缶、ストローなど）が記録されているため、実際の画像をいくつか目視で確認し、誤検出が頻発していないかを点検することが望ましい。たとえば、細長い植物をストローと誤検出している場合は、可視化時にカテゴリを制限する、あるいは除外処理を追加するといった方法で、地図上の情報をより正確に表現できるようになる。本システムでは、UI でクラス ID を指定することで、そのクラス ID に絞った分布を可視化した。

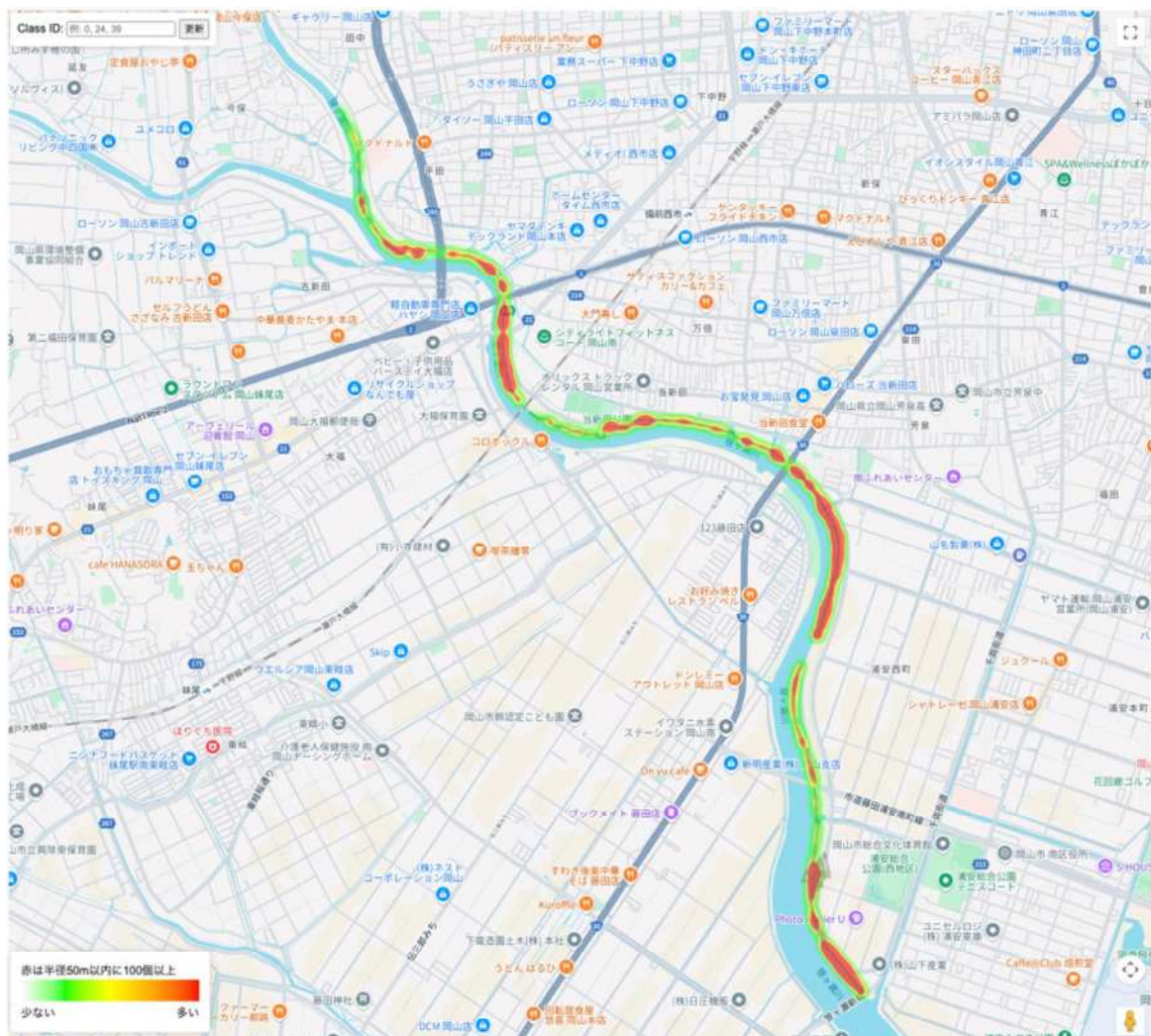


図4 全ごみ情報を使って作成したヒートマップ

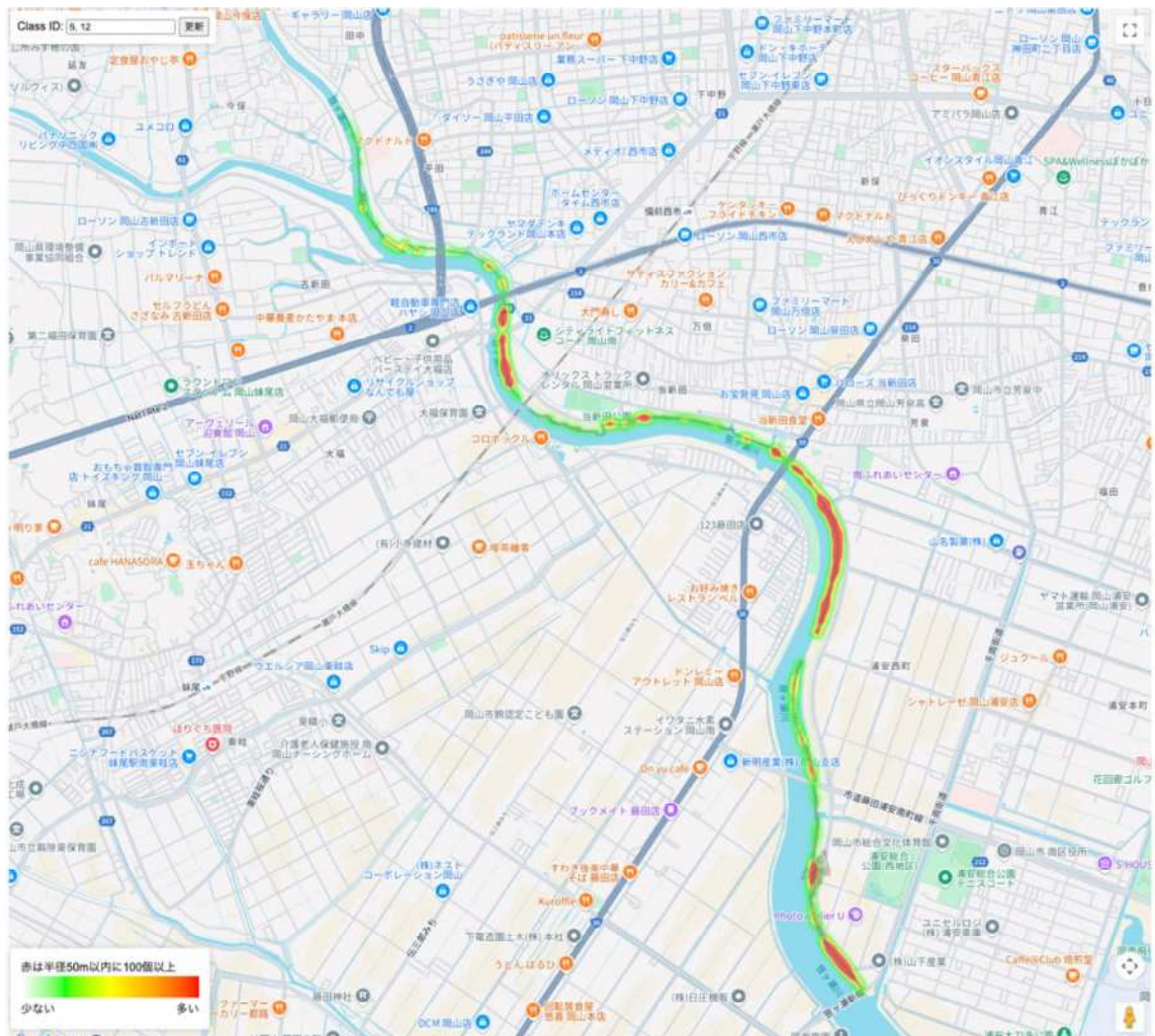


図5 ペットボトルと缶に絞った時のヒートマップ

3. 結果

3.1 ごみの検出精度

本プロジェクトでは、ごみの検出精度を確認するために、実際のデータを複数サンプルとして取り出し、目視で数えたごみの数とAIの検出結果を比較した。具体的には、無作為に選んだ10枚の画像（今回はファイル名の通し番号下二桁が01のもの10枚を選択している）を用いて、人の目で確認できたごみの数を「目視カウント」、AIが検出したごみの数を「AI検出数」として整理した（表3）。特徴的な結果としては、誤検出しやすいものは植物であり、検出漏れが起きやすいのは密集したごみであった。

表5 画像ごとに検出精度をまとめた表。実際の画像は文末を参照

画像名称	目視	モデルの結果	誤検出の数	検出漏れの数	誤検出した物	検出漏れていた物
B_B_右岸ーBー1 静止画 _DJI_0201	1	1	0	0		
B_B_右岸ーBー2 静止画 _DJI_0301	3	4	1	0	草	
B_B_右岸ーBー2 静止画 _DJI_0401	12	6	0	6		泥を被ったプラスチック、草陰のペットボトル
D_右岸ー静止画_DJI_0501	2	2	0	0		
E_右岸ー静止画_DJI_0601	5	4	0	1		密集した缶
F_右岸ー静止画_DJI_0701	26	19	0	7		密集した缶やプラスチックごみのいくつか
G_Gー2_右岸ー静止画 _DJI_0201	11	8	0	3		草に乗ったプラスチックごみ
G_Gー2_右岸ー静止画 _DJI_0301	2	1	0	1		沈んだ缶
G_Gー3_右岸ー静止画 _DJI_0401	11	6	0	5		密集したごみ
G_Gー4_右岸ー静止画 _DJI_0601	3	7	1	1	草	草の上のペットボトル

ごみの検出精度をより向上させるためには、現状で多く誤検出の原因になっている浮遊植物などを排除できるような対策が必要となる。 検出されたごみの中で、2 番目に多いカテゴリ Unlabeled litter と、Cigarette については、実際検出された画像を目視で確認した結果、植物の誤検出が多くみられた。そのため、ペットボトルや空き缶など典型的なごみの種類に学習データを絞ったり、密集したごみの画像を学習データに含めることで、精度が向上することが考えられる。現在のモデルでは河川に浮いている植物をごみとみなしてしまうケースが見られるため、次の段階では、こうした誤認を減らすための学習データの再構成や検出アルゴリズムの改良が重要な課題となる。

3.2 静止画と動画の比較

静止画と動画の比較に関しては、動画の方がフレーム単位で取り出せるためデータ量が多く、さらに検出時のログファイルとフライトログのマッチングという手順が増えるため、結果として解析に時間がかかるというデメリットがある。一方で、動画は連続的な映像を見せられるため、プロジェクトの内容を視覚的にアピールしやすく、宣伝素材としては非常に有効と考えられる。ただし、解析時間を短縮するためにフレームレートを下げると、実質的に一定間隔で撮影した静止画を扱うのとは変わらない点には留意が必要である。また、静止画の場合は撮影時の座標情報がヘッダーに保存されるが、動画ファイルにはそうしたメタデータが直接含まれていないことが多い。ごみの位置を正確にマッピングする必要があるユースケースでは、撮影座標とドローンカメラの姿勢が比較的容易に取得できる静止画を使う方が運用上の負担は小さい。

3.3 今後の課題

本システムは、河川のごみ分布を効率的に調査することを可能にし、従来の手法と比較して大幅な時間短縮と精度向上を実現した。しかしながら、現時点では依然として画像検出に取りこぼしや誤検出が発生しており、さらなる精度向上が求められる。特に、特定のカテゴリに対する識別精度が不十分なケースが散見されるため、より正確な検出を実現するには、対象カテゴリを限定した集

中的な学習が必要になる可能性がある。今後は、誤検出の傾向や取りこぼしが発生しやすい要因を分析し、それに基づいたデータ収集と学習の最適化を進めることが重要である。

4. 運用に向けたカスタマイズの提案

4.1 低解像度での検出

運用に向けたカスタマイズとして、まず 低解像度での検出 を検討することで、計算負荷を軽減し、より汎用的に広く活用できる可能性がある。低解像度画像を用いることで処理の高速化が期待できる一方で、現在のシステムのように小さなごみを検出するには不向きである。このため、低解像度での運用を考える場合には、ドローンを低空で飛行させるなど、撮影方法に工夫を加えることが重要となる。

4.2 検出するカテゴリの選定

次に、検出するカテゴリの選定 についても最適化を進める余地がある。現状は TACO dataset を用いたごみの識別を行なっているが、河川に存在する典型的なごみの種類は限られている（例：カン、ペットボトル、レジ袋など）。そのため、識別対象を限定し、少ないカテゴリで学習を行うことで、検出精度の向上が期待でき、より実運用に適したシステムとなる可能性がある。

4.3 学習用ドローンデータの拡充

また、学習用ドローンデータの拡充 も有効な手段の一つである。同一機種のドローンを使用して別の場所で撮影した画像を学習データに追加することで、検出モデルの適応力を高めることができる。これにより、異なる環境や撮影条件下でも高い検出精度を維持しやすくなり、より安定した運用が可能となる。

5. まとめ

本プロジェクトでは、物体検出 AI を活用した河川ごみ分布の効率的な調査を実施し、静止画解析を中心に高精度な検出結果を得ることができた。従来の目視確認に比べ、作業時間の大幅な短縮が可能であることが示され、清掃活動や環境保全計画におけるデータ活用の有効性が確認された。

特に、検出結果を地図上に可視化することで、ごみの分布傾向を一目で把握できる仕組みを構築した点は大きな成果である。また、解析したデータを基に、誤検出の原因として浮遊植物や密集したごみの識別が課題であることが明らかになった。これに対処するためには、今後さらなる学習データの拡充やモデルの改良が求められる。

一方で、動画データの解析については、フライトログとのマッチングの自動化に課題が残り、静止画に比べて処理コストが高いため、実運用に向けた最適化が必要であることがわかった。今回の試験では、バウンディングボックス付きの動画出力に留まったが、視覚的な啓発資料としての活用や、より詳細な時間経過を考慮した分析への応用が期待される。

今後、本手法を他の地域や異なる環境条件下でも適用し、検出精度のさらなる向上を目指すことで、環境保全活動の効率化に寄与できる可能性がある。本プロジェクトを通じて得られた知見を活かし、AI 技術を活用した環境保護の新たなアプローチを推進していくことが期待される。

Appendix 1 検出された画像例



図6 B_B_右岸-B-1 静止画_DJI_0201.JPG



図7 B_B_右岸-B-2 静止画_DJI_0301.JPG



图8 B_B_右岸-B-2 静止画_DJI_0401.JPG



图9 D_右岸一静止画_DJI_0501



图10 E_右岸—静止画_DJI_0601.JPG



图11 F_右岸—静止画_DJI_0701.JPG



图12 G_G-2_右岸一静止画_DJI_0201.JPG



图13 G_G-2_右岸一静止画_DJI_0301.JPG



图14 G_G-3_右岸-静止画_DJI_0401.JPG



图15 G_G-4_右岸-静止画_DJI_0601.JPG

Appendix 2 解析手順

今回の解析を Google Colab 上で再現するための手順を示す。

ごみの検出

学習済みモデルも含んだ Google Drive フォルダをコピーし、Trash_Detection.ipynb を開く（下記 URL）

https://drive.google.com/drive/folders/1KxleQdmNYLVoVmvqkVK8DVneS7XVPfQ9?usp=drive_link

1. ごみの物体検出対象となる画像、学習済みのモデルを Google Drive の作業ディレクトリに保存する。（デフォルトではコピーした Google Drive のディレクトリ構造に則っている。）
2. ランタイム—ランタイムタイプを変更 T4GPU を選択
3. モデル、物体認識対象の画像や動画、バウンディングボックス付きでアウトプットする画像や動画、ログファイル、それぞれのパスを設定する。
4. 設定用のセル（一番上のセル）を実行してから、コメントに従って必要なセルを実行する。

注意点

- ・ 動画は特に処理が重くなりやすいので、フレームレートや画質を下げることを検討した方が良い。
- ・ 全ての画像を一気に実行するのも重くなりやすいので、画像を選別するか、性能の高い GPU を選択した方が良い。

ログファイルから、Google Maps 上にヒートマップを表示する

1. Google Maps API のアクセスキーを取得
 - a. Google Cloud Console にログイン。
 - b. 新しいプロジェクトを作成（または既存のプロジェクトを使用）
 - c. 左上の「プロジェクトの選択」をクリックし「新しいプロジェクトを作成」を選択。
 - d. プロジェクト名を入力し、作成。
 - e. 「Maps JavaScript API」を有効化
 - f. 「API とサービス」→「ライブラリ」に移動。
 - g. 「Maps JavaScript API」を検索して有効化。
 - h. API キーを取得
 - i. 「API とサービス」→「認証情報」に移動。
 - j. 「+ 認証情報を作成」→「API キー」をクリック。

- k. 作成された API キー をコピー。(後から使う)
2. 必要なファイルを、作業ディレクトリの下に準備
github リンク : <https://github.com/Solafune-Inc/TRASHDETECT>
 - a. └── heatmap.html (ヒートマップを表示する HTML)
 - b. └── detected_images_log.csv (ログファイル: 緯度・経度データ)
3. heatmap.html の作成。このリンクをコピー
4. ローカルサーバーを立ててヒートマップを表示(Python 環境が必要)
 - a. ターミナル (コマンドプロンプト) を開く
 - b. heatmap_project/ フォルダに移動
 - c. ローカルサーバーを起動
`python3 -m http.server 8000`
 - d. ブラウザでヒートマップを開く
<http://localhost:8000/heatmap.html>
5. Class ID フィルタを使う
 - a. ページ上部の入力ボックスに 0, 24, 39 などの Class ID を入力して「更新」ボタンを押す。
 - b. その Class ID に該当するデータのみがヒートマップに表示される。
 - c. 空欄のまま「更新」するとすべてのデータが表示される。

ログファイルから、geopandas を使用してヒートマップを表示する

1. ログファイルを作成後、make_heatmap.ipynb を上から実行する
2. ヒートマップが表示される html ファイルが作成されるので、ブラウザで表示



モデルの追加学習

この URL を自身の環境にコピーし、上から実行

https://colab.research.google.com/drive/1cxAgQ-Hm97S_d5ribkv5G7kc_-lGq_Df?usp=sharing

- ・ TACO dataset と YOLO ではアノテーションの形式が異なるので、YOLO 形式に変換し、学習を進めている。
- ・ ディレクトリの構造が正しいことが重要。