SIGNATE 漁業×AIチャレンジ
魚群検知アルゴリズムの作成（通称鰹節コンペ）
2nd Place Solution

@yu4u

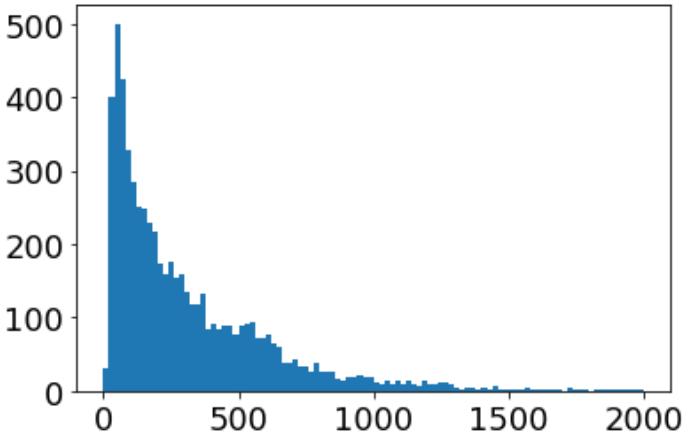
コンペ概要

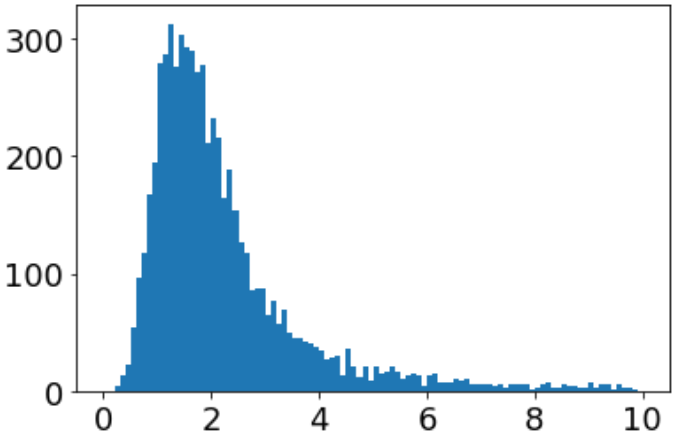
- 2クラス物体検出タスク
  - Jumper School
    - 「白湧き群」魚が海面上へ跳ねることで生じる飛沫が白く見える状態
  - Breezer School
    - 「水持ち群」海面直下にいる魚の動きが風波を打ち消してできる海面の凪
- データセット
  - ドローンの空撮動画から1秒単位でフレームを切り取り生成したもの
  - train: 3387枚, test: 1481枚
  - 画像サイズ：3840x2160
  - train, testともに動画ごとに系列情報が与えられている
- 評価指標
  - MAP@IoU=**0.3**

バウンディングボックスサイズ等

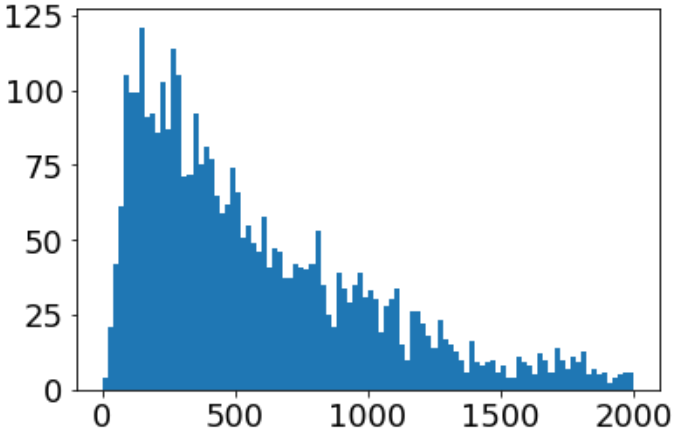
サイズ (sqrt(w*h))　　アスペクト比 (w/h)

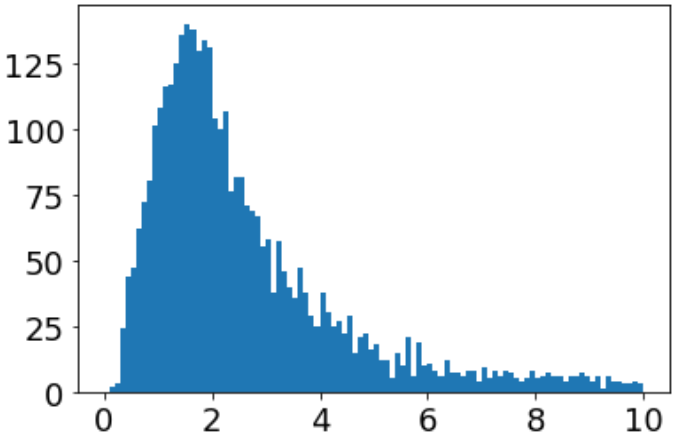
白湧き群





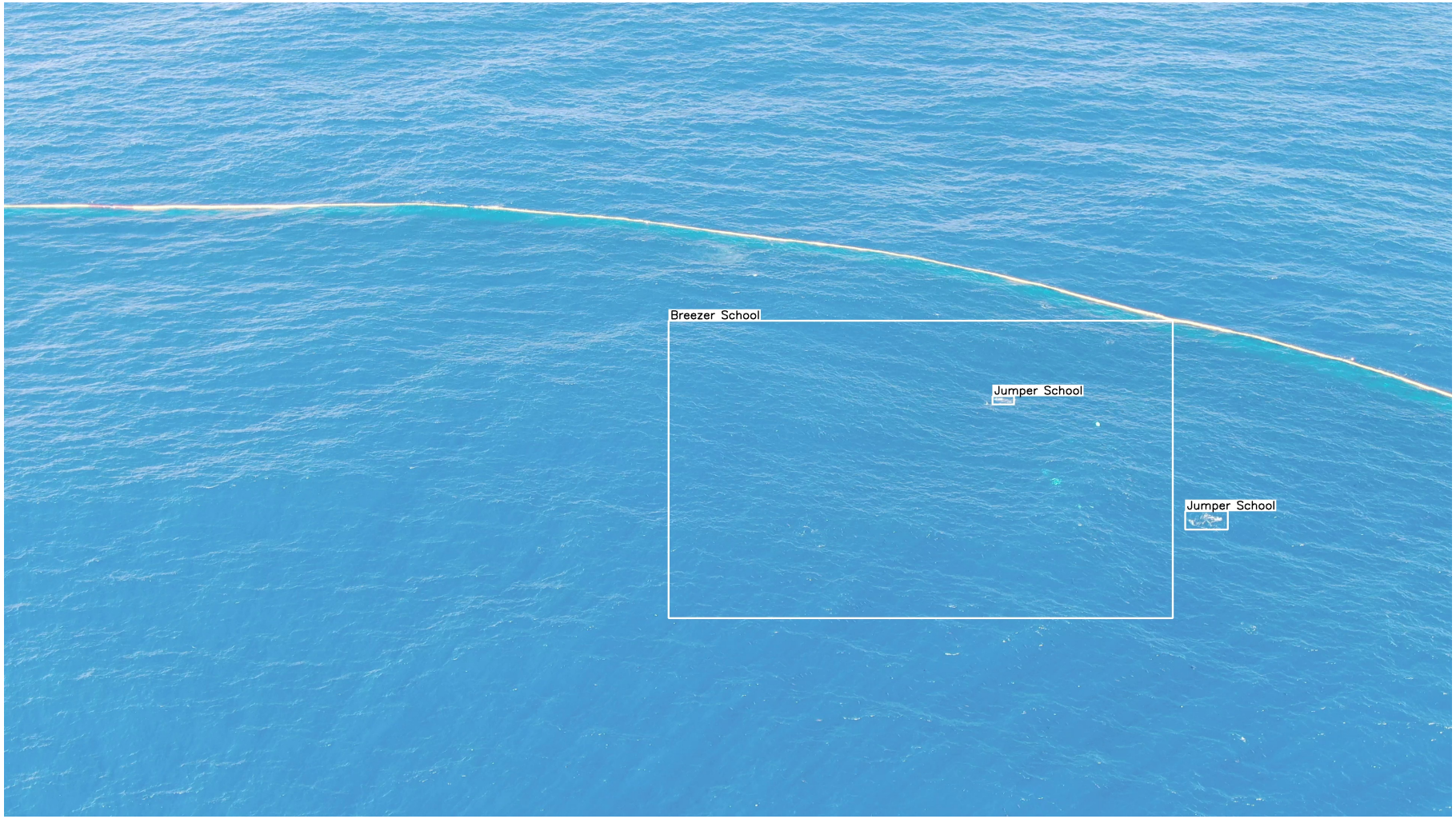
水持ち群



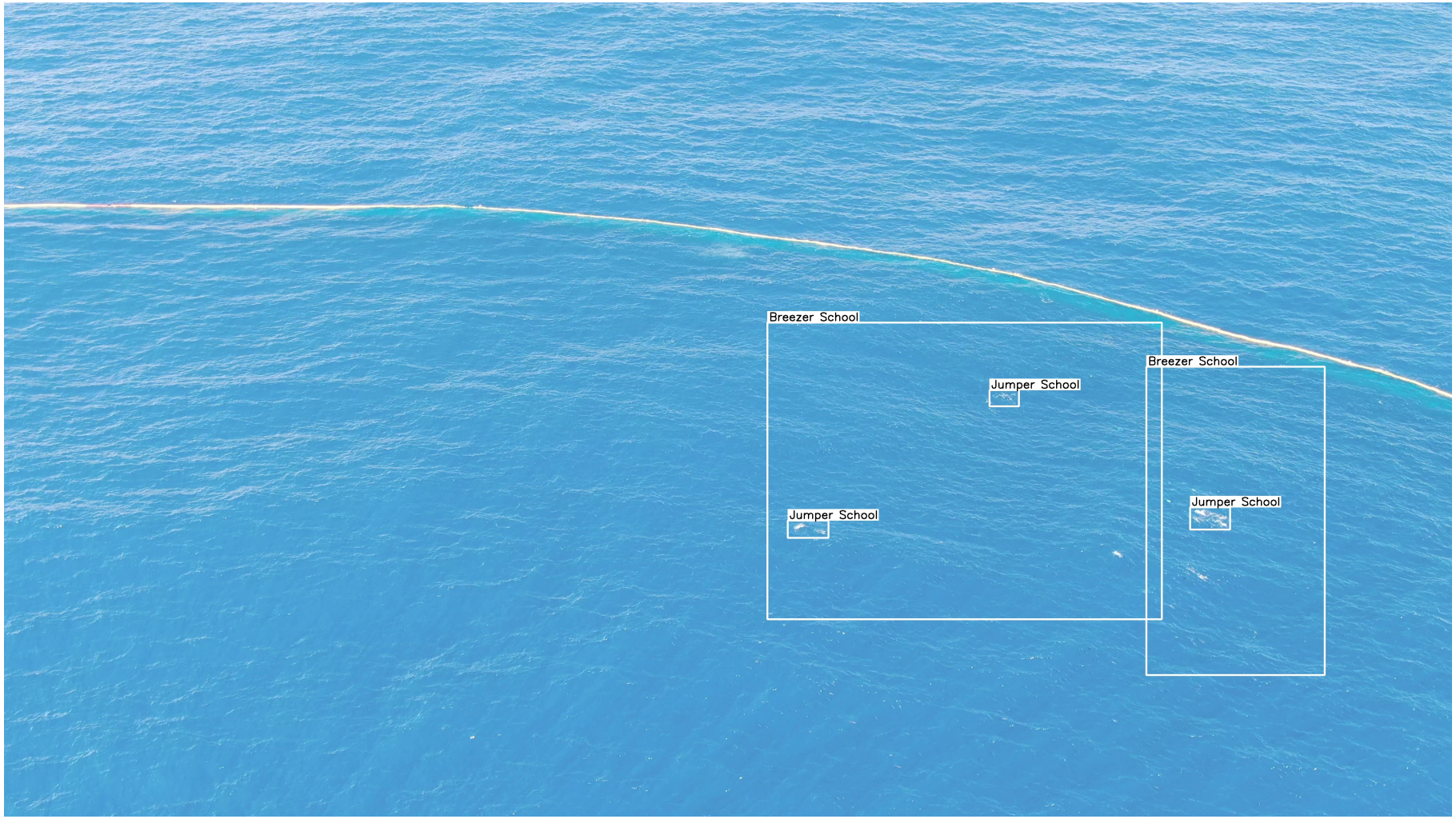


- 白湧き群は小さいBBOXが多い；BBOXはどちらも横長

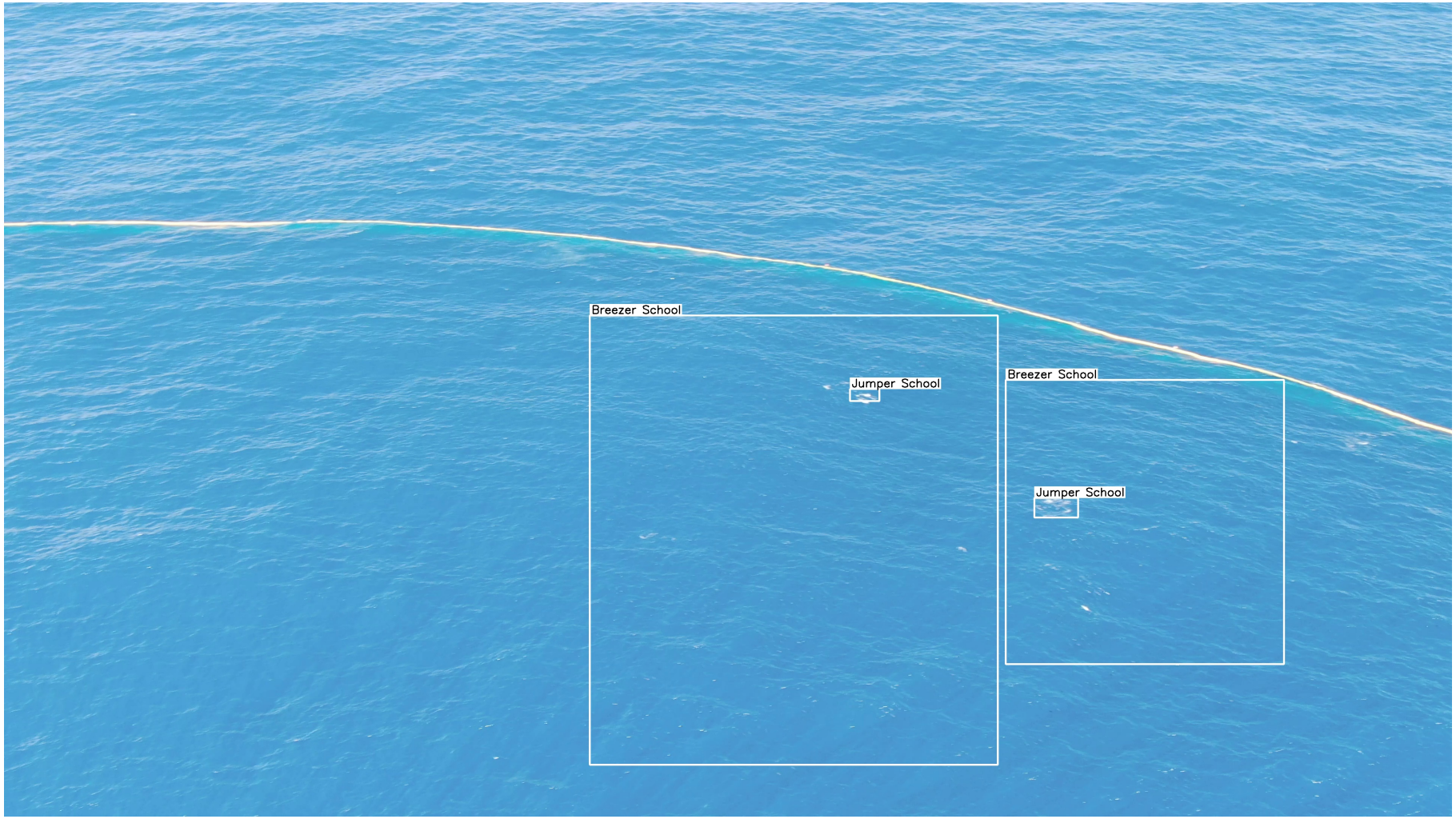
アノテーション例



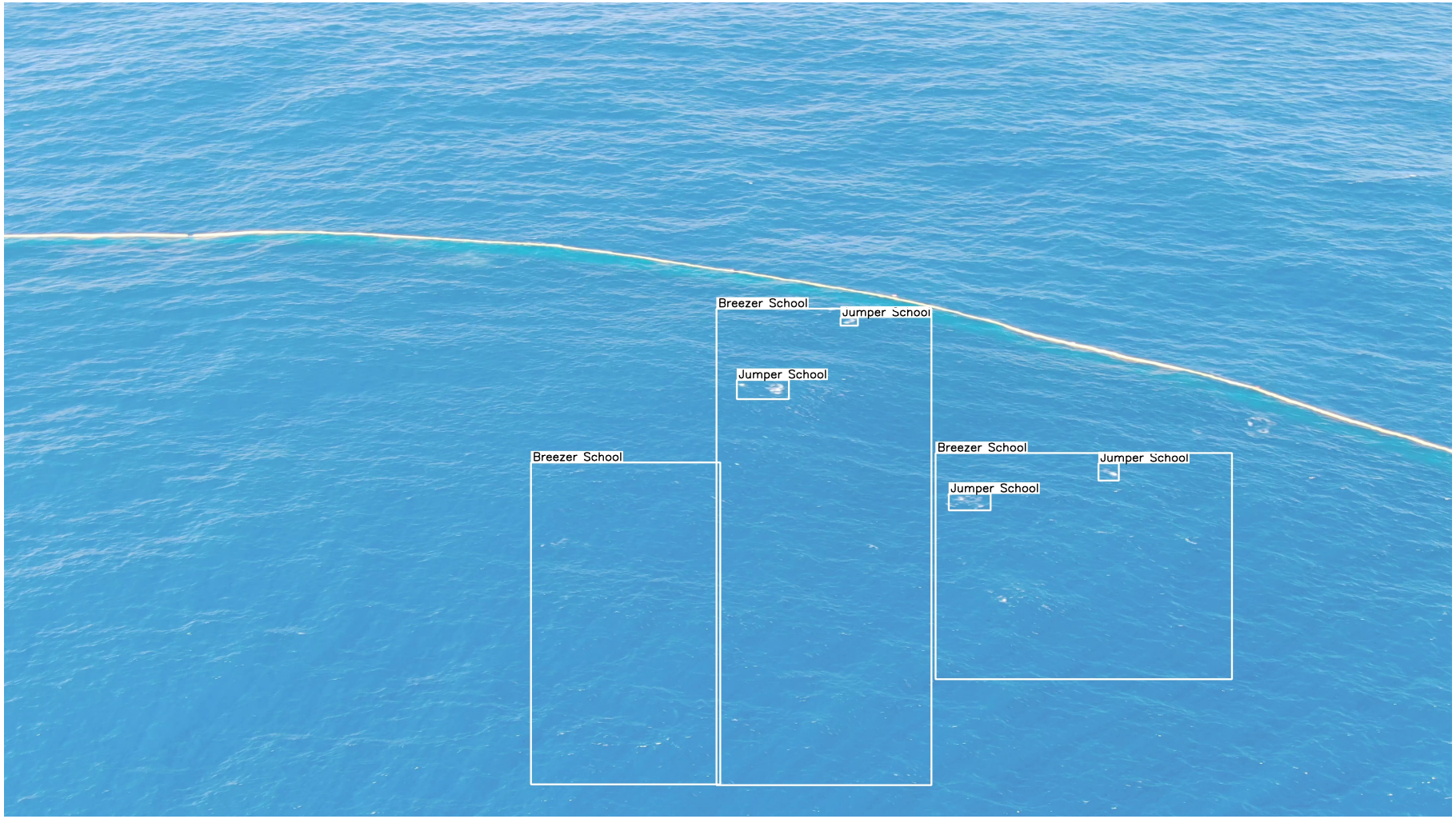
アノテーション例


Breezer School
Jumper School
Jumper School
Breezer School
Jumper School

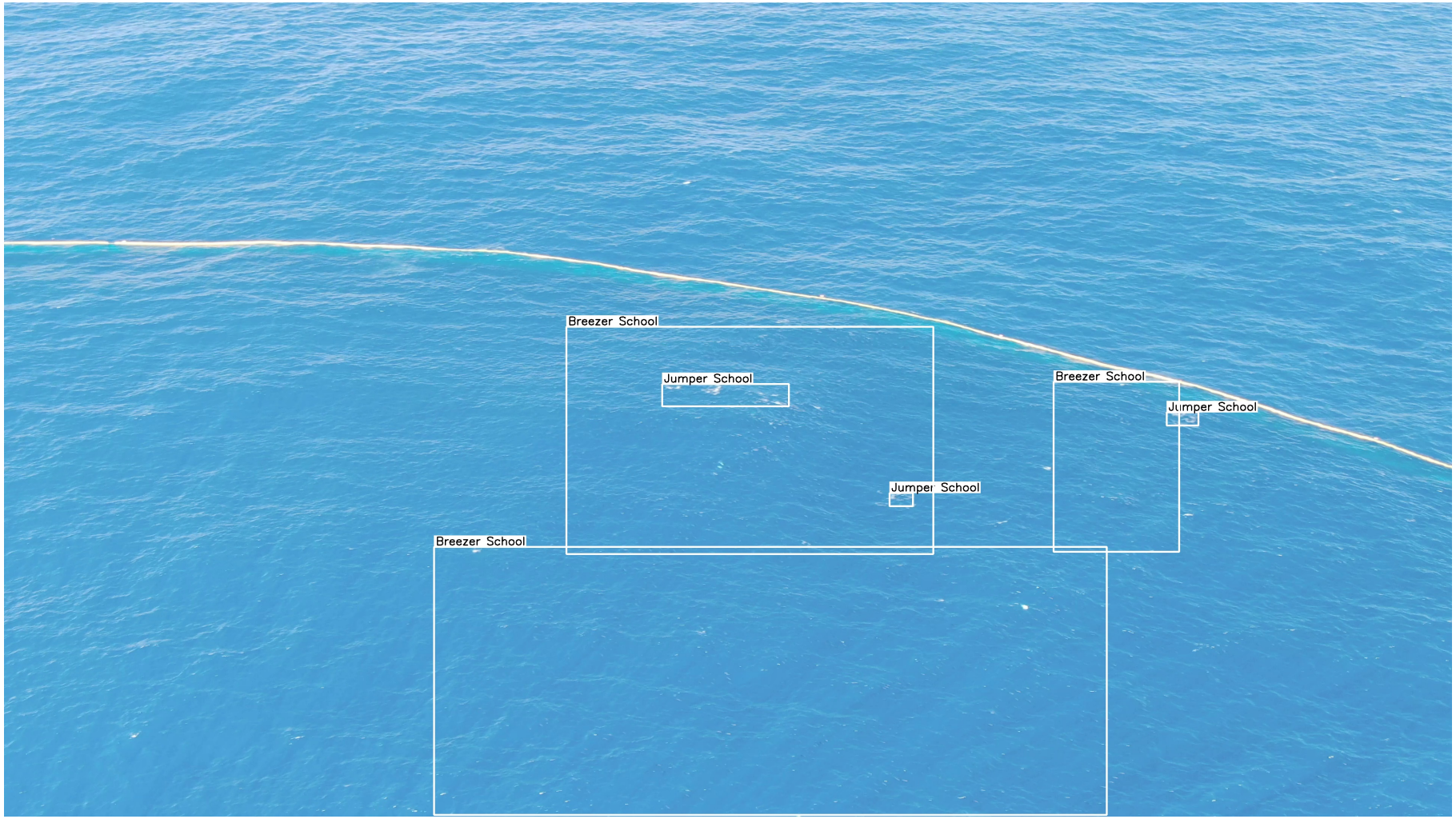
アノテーション例


Breezer School
Jumper School
Breezer School
Jumper School

アノテーション例


Breezer School
Jumper School
Jumper School
Breezer School
Breezer School
Jumper School
Jumper School

アノテーション例


Breezer School
Jumper School
Breezer School
Jumper School
Jumper School
Breezer School

データの特性

- 非常に小さなBBOXから大きなBBOXまで存在
  - 画像サイズが大きいが小さくリサイズしすぎては駄目
- 水面に対して平行に近い角度もあり、横長BBOXが多い
- アノテーションが恐らく時系列的にバラバラに作成されており
  ゆらぎが存在する
- 上記のように正解が定義しづらいことから検出IoU閾値が緩め（0.3）

→ 個々のモデルを頑張るよりアンサンブルでなんとかする方針

アプローチ

- データsplit
  - 5-fold cross validation (GroupKFold)
    - Sequence idによりgrouping
- モデル
  - EfficientDet 2パターン ＊ 5
  - YOLOv5 2パターン ＊ 5
- アンサンブル
  - Weighted boxes fusion (WBF) で統合
  - ハイパラをOptunaで最適化
- 時系列情報は利用せず
- No TTA

EfficientDetモデル

- EfficientDet: EfficientNetバックボーン＋BiFPNのsingle shot detector
- 実装 https://github.com/rwightman/efficientdet-pytorch
- バックボーンはefficientnet-b3のみを利用
- Augmentation
  - Resize (1920, 1024), RandomCrop, RandomRotate90, HorizontalFlip, RandomBrightnessContrast, ShiftScaleRotate
  - Crop size (1024, 1024) と (768, 768) の2パターンのモデル
    - **effdet_b3_1024**, **effdet_b3_768**（どちらもbatch size 4）
    - 本当はResizeで解像度を変更したモデルとする予定だったがミスっている
- Optimizer
  - AdamW, lr: 1e-4 -> 1e-5 (CosineAnnealingWarmRestarts), 30 epochs
  - Automatic Mixed Precision利用

YOLOv5モデル

- YOLOv5: 色々頑張ってるオレオレYOLO実装フレームワーク
  - s, m, l, xの4種類のバックボーン（depth, widthが違う）
- 実装 https://github.com/ultralytics/yolov5
- 画像を全て (1024, 1024) にリサイズし、上記repoをそのまま利用
- YOLOv5x, YOLOv5lの2パターン
  - **yolov5x** (batch size 8), **yolov5l** (batch size 14)

アンサンブル

- 4タイプのモデル
  - effdet_b3_1024, effdet_b3_768, yolov5x, yolov5l
  - 5-fold → 20モデル
- テスト時は20モデルの検出結果をweighted boxes fusionで統合
  - IoUが閾値より大きいBBOXをまとめ座標とスコアをアップデート
  - https://github.com/ZFTurbo/Weighted-Boxes-Fusion
- ハイパラはOptunaで最適化
  - 4タイプのモデルconfidence係数（effdet_b3_1024は1固定）
  - IoU閾値
    をOptunaを利用しoof（train全体）予測値で最適化
  - → **w1**: 0.5189, **w2**: 1.5534, **w3**: 1.7587, **iou_thr**: 0.2799
    - iou_thrのデフォルトは0.55。YOLOv5はeffdetと比較して低conf

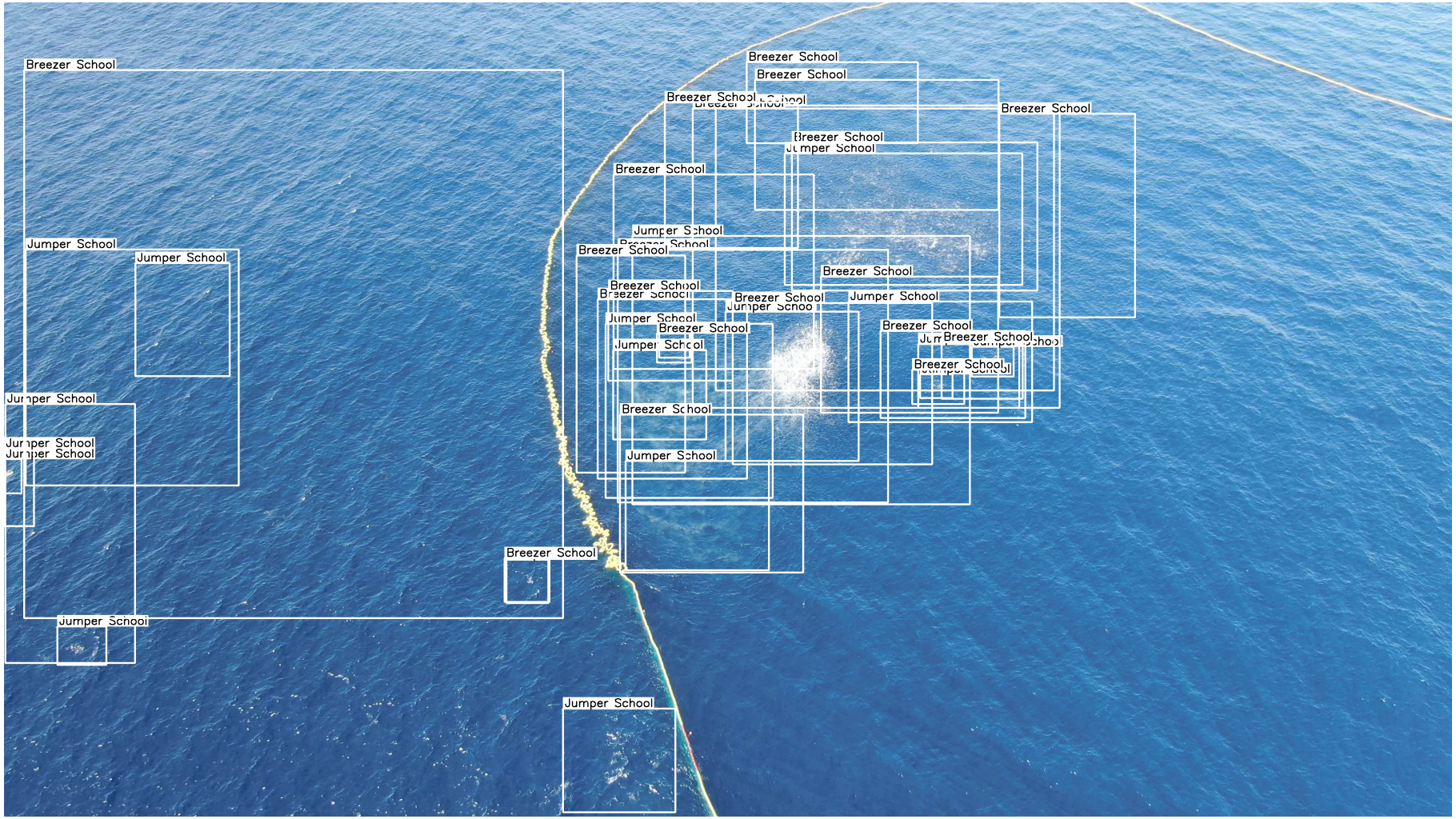
細かい点

- README: “1評価画像に対する各カテゴリの最大予測領域数は20とする”
- それぞれちゃんと20まで予測値を入れるため、detectorのconfidenceしきい値を大きくしすぎない
  - cf. “Evaluating Large-Vocabulary Object Detectors: The Devil is in the Details” https://arxiv.org/pdf/2102.01066.pdf

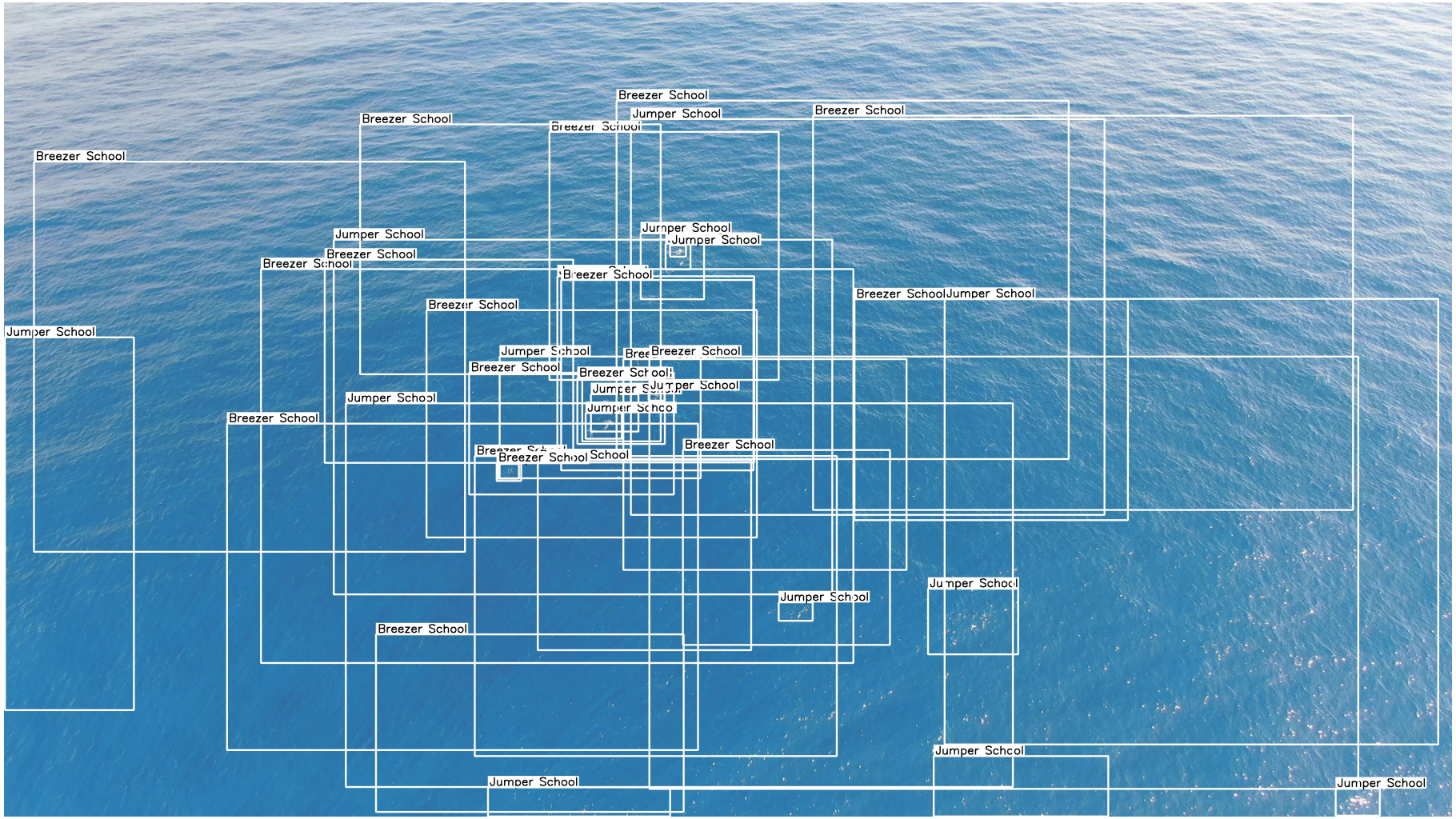
結果

	CV	暫定評価	最終評価
effdet_b3_1024	0.49171		
effdet_b3_768	0.45323		
yolov5x	0.44918		
yolov5l	0.44390		
ensemble (default)	0.55192		
ensemble (opt IoU)	0.56624		
ensemble (opt all)	0.58444	0.6973349 (1st)	0.6950604 (2nd)

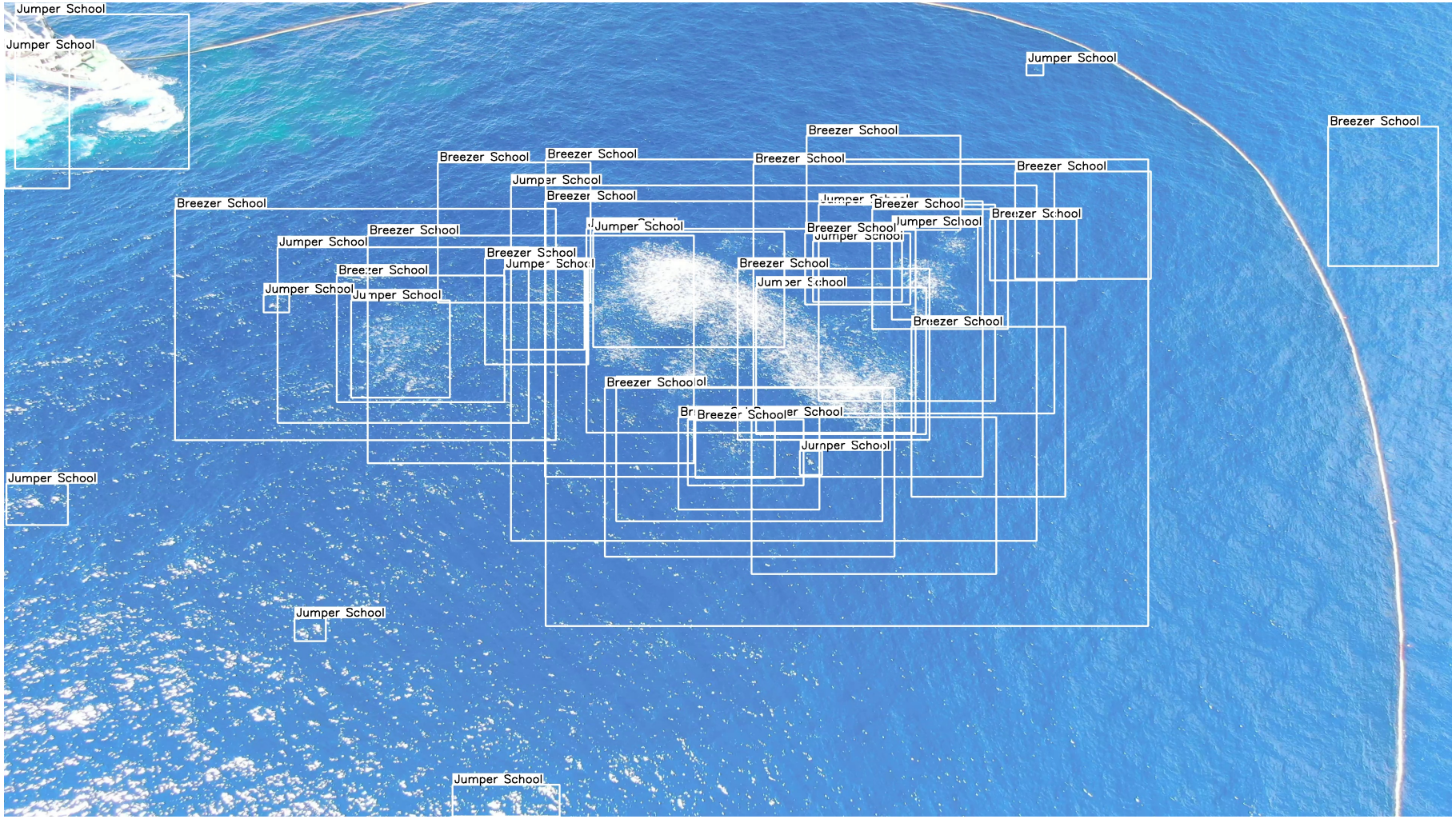
投稿ファイル可視化例



投稿ファイル可視化例



投稿ファイル可視化例


Jumper School
Jumper School
Jumper School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer School
Jumper School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer School
Breezer

まとめ

- 手法サマリ
  - EfficientDetとYOLOv5のアンサンブル
  - アノテーションにゆらぎがあることと、0.3という低IoU閾値にアンサンブルのハイパラ調整がうまく効いた（多分）
- やりのこしたこと
  - 時系列情報の活用

利用フレームワーク等

- パイプライン: pytorch-lightning
- config管理: OmegaConf
- 実験管理: wandb
- EfficientDet: https://github.com/rwightman/efficientdet-pytorch
- YOLOv5: https://github.com/ultralytics/yolov5
- Augmentation: albumentations
- BBOXアンサンブル: https://github.com/ZFTurbo/Weighted-Boxes-Fusion
- ハイパラ調整: Optuna