

疑似ラベルを用いた自動運転のための遠赤外線画像からの物体検出

B4 加藤 達也

1 研究背景および目的

- 背景： 完全自動運転の実用化に向けて技術の開発が進められており、その為に車載カメラ画像からの物体検出は重要な要素技術である。可視光画像からの物体検出は天候や時間帯によって精度が低下するので、その解決策として遠赤外線からの物体検出手法を考える。
- 課題:遠赤外線画像のデータセットは可視光画像のデータセットと比較して数が少ない。
- 目的： 遠赤外線画像を入力として低照度下でも安定的に動作する検出モデルを構築する。また、RGB 画像に適応して得た検出領域を教師とするドメイン適応を用いて、遠赤外線領域における検出モデルを構築する。



図 1: RGB 画像



図 2: FIR 画像

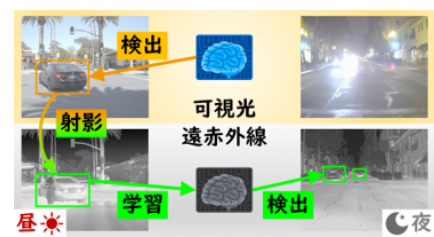


図 3: ドメイン適応の流れ

2 これまでの研究のまとめ

- 谷本先輩の最終的な提案手法は以下の通りである。流れとしては図 4(a) の通りである。
 1. RGB 画像に対して高精度な可視光モデルを用いて物体検出を行い、その結果を FIR 画像に変換して疑似ラベルを生成する。
 2. 可視光モデルの出力信頼度に基づき、高信頼の検出結果のみを疑似ラベルとして使用し、誤学習を防止する。
 3. FIR 画像と疑似ラベルを用いて、可視光モデルをファインチューニングし、FIR 画像専用の物体検出モデルを構築する。
- 損失関数のカスタマイズを行った。採用した損失関数は FocalLoss と L1Loss。
- クラス分類とオブジェクト性に FocalLoss、バウンディングボックス回帰に SmoothL1Loss を使用した。
- 現在、サイズが小さいオブジェクトに対しての精度が低いため、難しいサンプルに着目し、分類性能を向上させる FocalLoss を採用した。
- SmoothL1Loss に関しては YOLOX で中心座標と幅、高さの回帰に対して L1 距離を使うことが多いため、採用した。

Category	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l
person	0.039	0.194	0.001	0.039	0.079	0.087
car	0.163	0.490	0.055	0.039	0.322	0.644

表 1: 各カテゴリの検出精度 (例)

- mAP は閾値 0.50 0.95 における検出精度の平均値である。

3 前回の GM からの進捗

3.1 損失関数の実装について

- config ファイルは base ファイルに上書きや追加をすることによってカスタマイズをする構造になっているが、改めて base ファイルの方を参照して見たところ、既に損失関数は実装されていた。
- bbox には IoULoss、クラス分類とオブジェクト性に CrossEntropyLoss が使用されていた。
- この 2 つとも、Yolox に最初から実装されている損失関数であったため、デフォルトの損失関数から他のカスタマイズされた損失関数に変更することによってどのように検出精度が向上するかを考えるべきであると思った。

3.2 谷本先輩の修士論文時点での検出精度と現時点での検出精度の比較

表 2: 谷本先輩の検出精度 ($\theta_{\text{det}} = 0.50$)

category	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l
person	0.024	0.108	0.001	0.027	0.072	0.100
car	0.299	0.567	0.293	0.119	0.553	0.758

表 3: 私の検出精度 ($\theta_{\text{det}} = 0.50$)

category	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l
person	0.370	0.771	0.300	0.245	0.452	0.601
car	0.304	0.647	0.238	0.116	0.401	0.649

- ($\theta_{\text{det}} = 0.50$) は信頼度であり、0.50 以上のものが疑似ラベルとして使用される。
- 手順は全く変わっておらず、疑似ラベルを作成→SAM2 でのアノテーション→学習・検証である。違いとしては損失関数の変更と、細かいパラメータの変更である。
- mAP に関しては person、car とともに越えている。ただ、car に関しては一部下がっている部分が存在する。
- 考えられる原因としては、クラス分類の損失関数として設定している FocalLoss において、バランス係数 (alpha) を person と car それぞれに 0.69 と 0.31 と設定してクラス不均衡を減そうとしたことである。

$$FL(P_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (1)$$

- (1) は FocalLoss の数式であり、 α はバランス係数であり、クラス間に不均衡がある場合に少ないクラスに対して損失の重みを大きくするために使う。
- γ は困難サンプルに集中するためのパラメータであり、判別が容易であるサンプルの影響を抑え、難しいサンプルに注目するための関数である。
- どちらも大きい数値であればあるほど影響は大きい。
- またバッチサイズが Yolox_s と Yolox_x でそれぞれ 16 と 8 であるが、学習率が一律で 0.01 になっていたので修正した。
- Yolox_s は 0.0025、Yolox_x は 0.00125 とした。
- 根拠としては YOLOX: Exceeding YOLO Series in 2021 において (2) で学習率を定めていたことからである。[1]

$$lr_{\text{actually}} = lr \times \text{BatchSize}/64 (lr = 0.01) \quad (2)$$

3.3 ITS ミーティングについて

- 愛知工科大学の久徳先生との毎週月曜日に行われる ITS ミーティングにおいて、アドバイスをいただいた。
- 使用されるクラス数に関しては、RGB 画像学習時、また疑似ラベル作成時は 80 クラス。学習・検証時は 2 クラスとする。
- 損失関数については既存の損失関数はそれぞれ想定された使い方が存在するので、それに従って適した損失関数を各所の実装する。
- 既存の損失関数でも良いが、谷本先輩は独自の計算での損失関数の作成を目指していた。

4 今後の課題&スケジュール

- FLIR のバージョンアップをする。
- 損失関数に関する理解を深め、より良い検出精度を目指す。できれば独自の計算で損失関数を作成したい。

参考文献

- [1] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun, YOLOX: Exceeding YOLO Series in 2021 (6 Aug 2021)