

道路標識検出 AI のための Unity を用いた アノテーションデータの自動化に関する研究

山崎 太郎 山富 龍 マハブービ シェヘラザード 二宮 洋
湘南工科大学工学部情報工学科

1. はじめに

原付以上運転者による交通事故件数は年々減少している。しかし、依然として総件数は多く、その7割以上が安全運転義務に反しているためと判明している。このような事故を防止する技術として、運転支援システムがある。このひとつに、車載カメラから得た映像を利用し、人や車、標識などの周囲の状況を推定する物体検出 AI を用いたものがある。しかし、高精度な物体検出 AI の実装には、学習データとなる実世界の画像とその画像に対して物体の位置と種類をラベル付けたアノテーションデータが多量に必要となる。実世界で様々な角度や大きさで映る目標物の画像データを集め、それに対しアノテーションを付与することは多量な手間と時間を要する作業となる。そこで、本研究ではこの問題を解決するため、Unity[1]の3D空間におけるオブジェクトに対して自動的に付与されるアノテーションに着目し、物体検出 AI の学習用データに対する応用を試みる。本研究では、実世界と Unity の 3D 空間内の道路標識に対して物体検出 AI を学習させ、比較を行うことで提案手法の性能を評価する。

2. 従来のアノテーション法

物体検出におけるアノテーションとは注釈や注解を意味し、画像や動画内の物体に対するタグや位置情報を記したメタデータである。物体検出 AI に使用される従来のアノテーション作業では、画像データ中にある目標物の周りを矩形領域で囲い、それに物体名などを付与する単純な作業である。高性能な AI の実装には多量の学習データが必要であり、そのためアノテーション作業の量と時間が増加する問題点がある。

3. Unity におけるアノテーションの自動化

本研究では、Unity に配置した 3D 都市モデル[2]、標識、カメラを用いて実世界に準じた環境、シーンを再現し、多量の道路標識が映る画像データと矩形領域が自動的に付与されたアノテーションデータの作成方法を提案する。

Unity に配置したカメラで撮影した様子を画像データとして生成する。ここで実世界の画像収集方法を再現するために、カメラの位置を移動させる。具体的には、カメラと標識が一定の距離に近づくときと撮影することとする。このとき、カメラと標識の距離と標識の大きさから、撮影した画像サイズに適した標識のある座標と大きさに変換し、標識が含まれている矩形領域が付与されたアノテーションデータを自動作成することができる。

4. 実験

4.1 実験環境

本実験では、Unity の学習データを用いて学習した物体検出 AI の性能を調査するため、実世界のデータにより学習した AI と比較を行う。検出対象の道路標識として駐車禁止を選択した。物体検出ネットワークは SSD[3]、学習アルゴリズムは SGD を用い、最大エポック数は 500、ミニバッチサイズは 32 に設定した。本実験では、CPU は AMD Ryzen 5 3600、メインメモリは 16GB、GPU は Nvidia GeForce RTX 3060 の計算機を使用した。

4.2 評価指標

本研究では、データの作成時間と検出の精度に対して比較と評価を行う。評価の指標として数値が 1 に近いほど高性能であることを示す AP(Average Precision)を用いる。画像一枚ごとに検出器の出力結果から、検出器の検出した物体が正しく検出できた割合を示す適合率(Precision)と設定した物体を検出器が検出できた割合を示す再現率(Recall)を算出する。Precision と Recall の計算式をそれぞれ(1)と(2)に示す。AP は、求めた再現率を昇順に並べ、再現率の増加分と適合率の値を掛け合わせ加算することで求める。AP の計算式を(3)に示す。

A Study of Automatic Annotation Data Creation Using
Unity for Road Sign Detection.
Taro Yamasaki, Ryo Yamatomi, Shahrzad Mahboubi,
Hiroshi Ninomiya
Department of Information Science,
Shonan Institute of Technology

$$Precision_i = TP_i / (TP_i + FP_i) \quad (1)$$

$$Recall_i = TP_i / (TP_i + FN_i) \quad (2)$$

$$AP = \sum_{j=1}^n (Recall_j - Recall_{j-1}) \times Precision_j \quad (3)$$

ここで、 i は画像の番号、 TP_i は正しく検出できた物体の数、 FP_i は誤検出した数、 FN_i は未検出の物体の数、そして n はテスト画像の枚数である。

4.3 Unity を用いた学習データの作成

3D 都市モデル[2]を用いて実世界に近い画像データを作成するため、太陽光や天候を追加する。太陽光を平行光源として3D都市モデルに配置し、実世界の影の変化を再現するために平行光源を0から180度を5等分した角度に設定し変化させた。天候を3D都市モデルの空模様を用意した晴れ3種、夕暮れ1種、曇り1種、夜1種の計6種類のテクスチャを変更することで再現する。

本実験では、検出対象の道路標識として駐車禁止を利用することとし、一枚の画像に複数の道路標識が映り込むシーンを再現するため、駐車禁止以外に駐停車禁止、指定方向進入禁止、追い越し禁止、展開禁止、速度制限50「km/h」の計6種類の道路標識を用意した。カメラが通る直線道路を30本用意した。その際、各道路を10分割し、うち2箇所複数の標識をランダムに設置することで様々なバリエーションのある画像を作成する。本研究で作成した実世界のデータとUnityを用いて作成したデータをそれぞれ図1.(a)および(b)に示す。



(a)実世界の画像 (b) Unity の画像
図1 収集した画像と生成した画像の比較

4.4 現実世界の学習データ

実世界で道路標識を含む画像を集めるためにドライブレコーダーを用いた。用意した合計517分の動画から駐車禁止標識が映る画像を600枚抽出し、手動で矩形領域を付与する作業を行った。

4.5 実験結果

本研究では、テストデータ用の現実画像を300枚用意し、実験を行った結果を図2に示す。図2では、縦軸をAP、横軸をエポック数としたときの

APの変化を示している。青線には実世界データを学習した検出器、赤線にはUnityデータを学習した検出器のエポック数に対するAPの変化である。

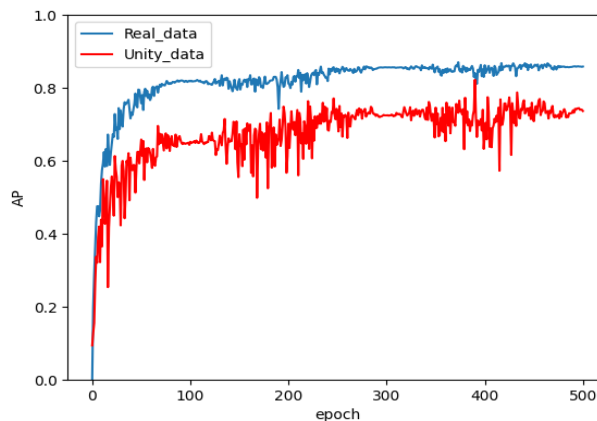


図2 エポック数に対するAPの変化

APの最大値は実世界データを学習した検出器が0.863であり、Unityデータを学習した検出器が0.823となった。結果からUnityを用いて作成した学習データが、実世界に即したシーンを再現できているため、実世界で収集した画像を学習した検出器のAPに近い検出精度を得られたと考えられる。

5. まとめ

本研究では、物体検出におけるアノテーション作業の簡略手法として、Unityを用いたアノテーションの自動化を提案した。実験結果より、Unityで作成したデータを用いて学習した検出器の精度が、実世界のデータを用いて学習した検出器に近い精度を得られた。今後の課題として、Unityに実装してない実世界にある様々な要素を追加し、Unityデータを学習した検出器の再現率を向上させることや、実世界では集めにくい悪天候を再現し、その有効性を調べる事が挙げられる。

参考文献

- [1]. Unity Technologies, “Unity”, <https://unity.com/ja> (最終アクセス:2022/10/05).
- [2]. 株式会社ゼンリン, “ZENRIN City Asset Series”, <https://www.zenrin.co.jp/contents/product/service/3d/asset/index.html> (最終アクセス:2022/11/13).
- [3]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Read, C.-Y. Fu, and A. C. Berg. “SSD : Single Shot MultiBox Detector”, *Proc. In ECCV*, pp.21-37. (2016).