

ディープラーニングによる物体検知技術 YOLOv5 を用いた図面要素抽出の研究

Table detection in highway completion drawings using an object-detection
algorithm YOLOv5

出口直輝¹⁾

Naoki Deguchi

銭谷誠司²⁾

Seiji Zenitani

名下宥佑³⁾

Yusuke Nashita

榎谷英弥⁴⁾

Hideya Masutani

概要：我々は地震、津波、大雨などの大規模な自然災害が都市にもたらす被害を予測するため、「都市丸ごとシミュレーション」を実現することを目指している。都市丸ごとシミュレーションは、バーチャル空間内で実際の3次元都市と対になる双子（デジタルツイン）を作ったうえで、この仮想都市を使って災害シミュレーションを行い、減災・防災に役立てるものである。都市デジタルツインデータを作るためには大規模構造物の設計図面データから情報を読み込む必要があり、そのための技術開発が必要である。本研究では、阪神高速道路株式会社が提供する高速道路の竣工図のスキャンデータから、構造物の情報を抽出することを目指し、その第一歩として図中の表要素を抽出することを試みた。ディープラーニングによる物体検知アルゴリズムYOLOv5を用いて竣工図から表を読み取る技術を開発し、その検出精度を評価した。

キーワード：都市丸ごとシミュレーション, デジタルツイン, ディープラーニング, YOLOv5

1. はじめに

近年発生している地震、津波、大雨などの災害は日本に大きな被害をもたらし、各地で被害者や死者が出ている。もはや日本には安全な場所は少なく、迫りくる災害に対してどのように対処し、被害を小さくするかが論点となっている。災害に対するため仮想都市空間(デジタルツイン)

を作成しその空間内で様々なシミュレーションを行う。そうすることで災害時の迅速な避難、被害規模想定などが容易になり、その都市の住民の安全性が高まる。その仮想都市空間を作るためには、大規模構造物から2次元の設計図を読み込み、バーチャル空間で3次元化する必要がある。その際、設計図面内にある構造物に関する構造物本体の情報や、使用されている部品の太さや材質に関するデータを読み込まなければならない。そのうえ、古い構造物の設計図はCADファイルではなく、人間が手書きした図面という形で保存されている。そこで、図面のスキャン画像（ラスターデータ）からデータを抽出する必要がある。

本研究では、設計図の2次元データ中に存在する表要素を抽出する。表に注目した理由は、図のメタ情報や、構造物の素材リストといった重要情報が表（テーブル）という形で図面に含まれているからである。図面の画像データを、ディープラーニングに基づく物体認識プログラムYOLOv5を使って解析して、図中の表を識別するところまでを担う。

2. 物体検知アルゴリズム YOLOv5

(1) YOLO の概略

YOLO (You Look Only Once) は、ディープラーニングに基づいた物体認識モデルの1つである。YOLOの特徴は物体を検知したい画像を正方形にリサイズした後、さらに細かく正方形で分割することであり、分析が容易になるとされている。¹⁾ あらかじめ正解の情報を与えた教師データから正解となる矩形(Box)の中心座標、幅、高さの情報から重みデータを作成する。その重みデータを使用して検知する画像から色情報などの特徴量から矩形を抽出するアルゴリズムである。検知される際、検知された部分が赤枠で囲まれ、その上部にどの物体が検知されたかという情報とその確信度が表示される。

YOLOは2022年1月現在、5つのシリーズが開発されている。2016年に発表されたYOLOv1を皮切りに、v2, v3, v4, v5と発表されている。しかし、YOLOはv1が開発された2016年以降、v3まではJoseph Rodmonが発表した以降、v4, v5は様々な研究者によって改変、発表されており、YOLOv5はv4から派生し、進化したと言えるかが今もなお、議論されている。²⁾ 本研究では最新シリーズであり、Glenn Jocherが2020年6月に公開したYOLOv5を使用する。

YOLOv5は処理が高速であり、リアルタイムでの実行も可能で、顔検知や自動運転などに応用されている。アノテーションデータを使って学習の重みデータを作成し、その重みデータを使って位置推定とクラス分類を行う。

(2) YOLOv5 モデルの実装

GitHubから以下のソースコードを参照した。

<https://github.com/ultralytics/yolov5>

YOLOv5のモデルには、ネットワークのサイズに応じて、s, m, l, xの4種類のモデルがある(図1)。最も小さなsモデルから最も大きなxモデルになるにつれて、必要なメモリ量が増え、学習に要する時間が増えるが、認識精度は高くなる。

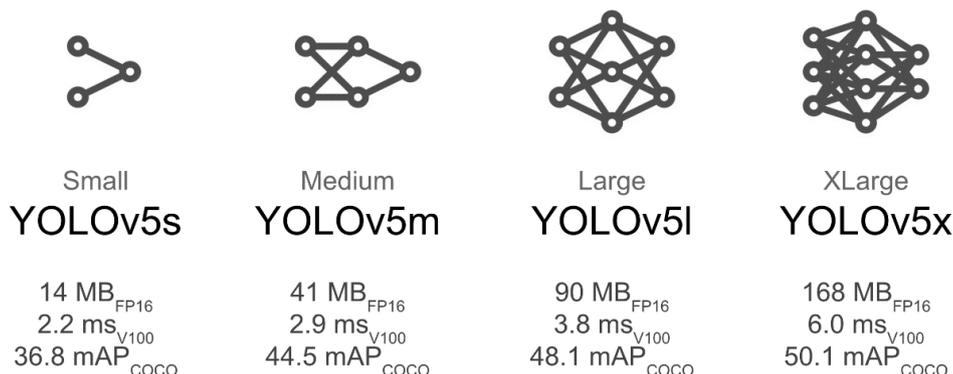


図1 YOLOv5 モデルの種類²⁾

(3) 損失関数・評価指標の実装

アノテーションデータの情報と YOLO モデルにより出力された検知範囲に関する情報の誤差が損失関数となり、この損失関数を小さくすることを目的とする。損失関数は YOLO 独自の形式となっている。

a) IoU(Intersection over Union)

IoU とは正解である範囲から出力した検知範囲からの近さを表す指標であり、図2 のよう共通部分から全体の面積を除して算出する。二つの領域が完全に一致している場合は IoU=1 となり、全く一致していなければ IoU=0 となる。

本研究ではデフォルト値である IoU \geq 0.45 となる矩形を TP(True Positive) とし、IoU<0.45 となる矩形を FP(False Positive) として学習を進めていく。

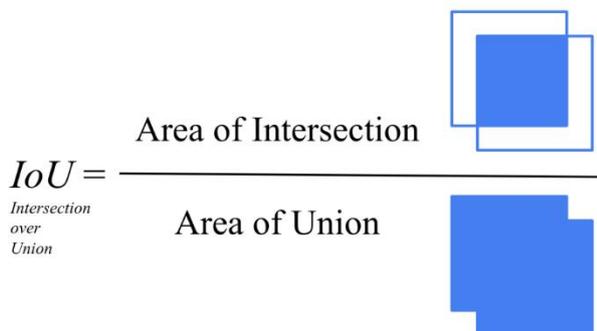


図2 Intersection over Union³⁾

b) 適合率(precision)

適合率とは、式(1)で表される、a)での TP, FP をもとに算出されるものであり、具体的には、真であると予想したときに真である確率のことを指す。

$$precision = \frac{TP}{TP + FP} \quad (1)$$

3. 教師データの作成

我々は、LabelImg (<https://github.com/tzutalin/labelImg>) という Python ツールを用いて、阪神高速道路株式会社が提供する高速道路の設計図(竣工図)のスキャンデータを作成した。

LabelImg のアノテート作業画面を図 3 に示す. 竣工図の画像ファイルを読み込んで, その中から識別したい領域 (図中の緑色の Box) と種類名を入力して, XML 形式のデータファイルに保存する. そして, この画像ファイルとアノテートファイルのペアを 9000 組用意して, ディープラーニングの教師データとして利用する.

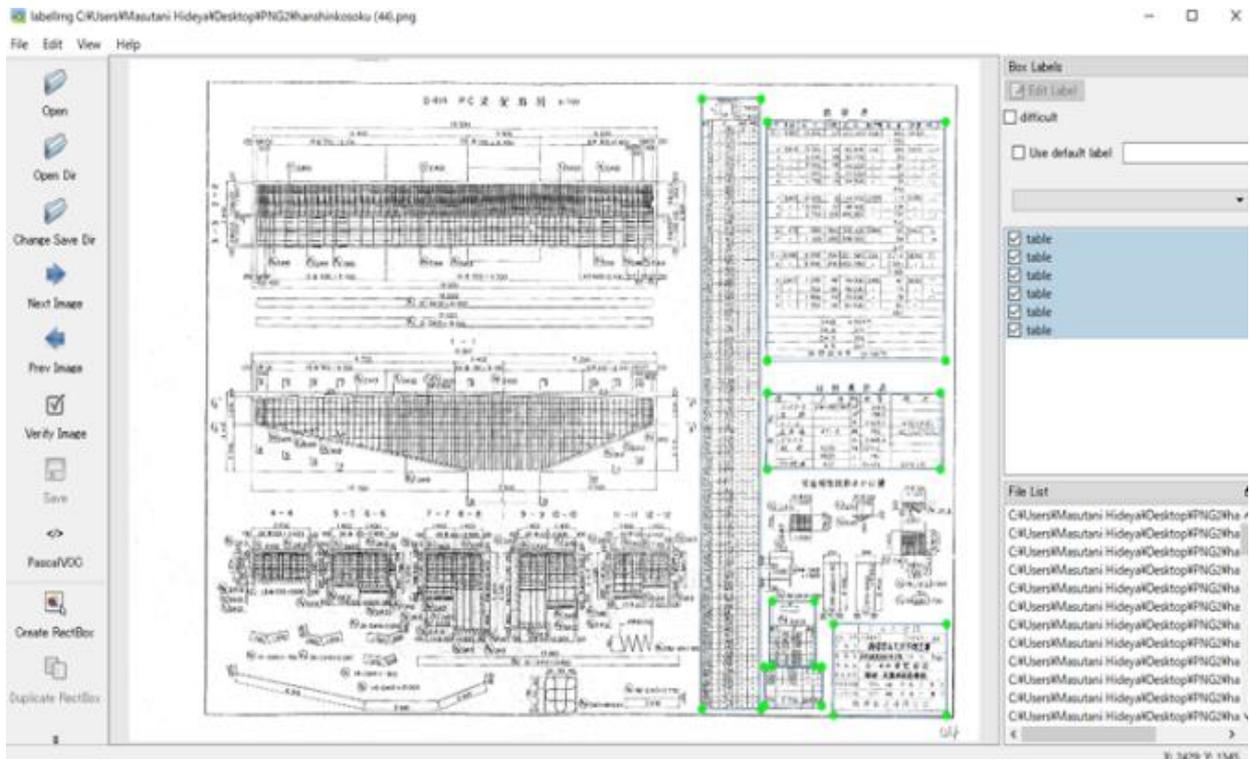


図 3 LabelImg の利用画面⁴⁾

4. 学習の実施

阪神高速道路の竣工図の画像とアノテートデータを読み込み, YOLOv5 を用いてニューラルネットワークを学習させた. 検出時に設定した確信度に関する閾値はすべて 0.7 としている. PC の環境は Windows10, GPU の環境は, NVIDIA GeForce RTX 3090, cuda. 11.1 を使用する

本研究では様々な条件を変更しながら学習を行った. まず初めに学習に使う教師データ数を変更した. 学習条件は学習回数(epoch 数)300 回, モデルは s モデルを使用し, 教師データ数を 100 枚, 300 枚, 1000 枚, 6000 枚としている. これらの枚数は学習用に使用し, それぞれ同じ枚数を評価用として扱う. しかし 6000 枚のケースでは準備した枚数では不足するため, 2000 枚を評価用とする. 推論に使う枚数は全ケースで 1000 枚を使用する.

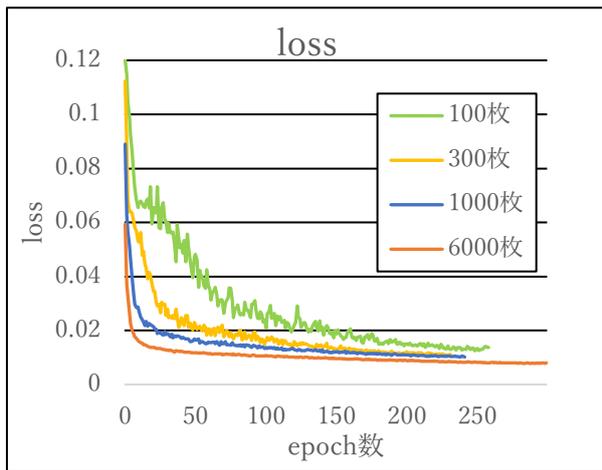


図4 枚数による loss の比較

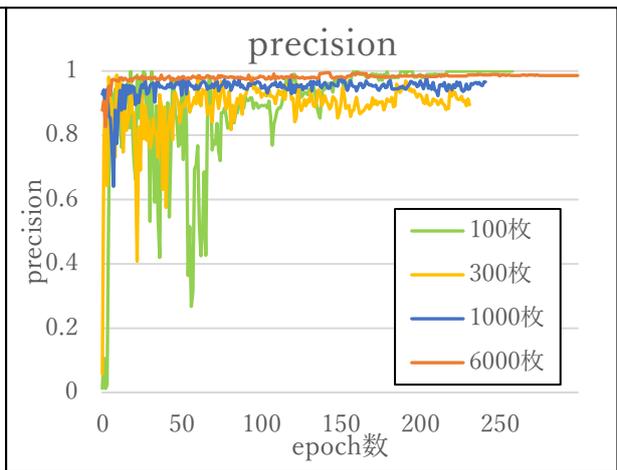


図5 枚数による precision の比較

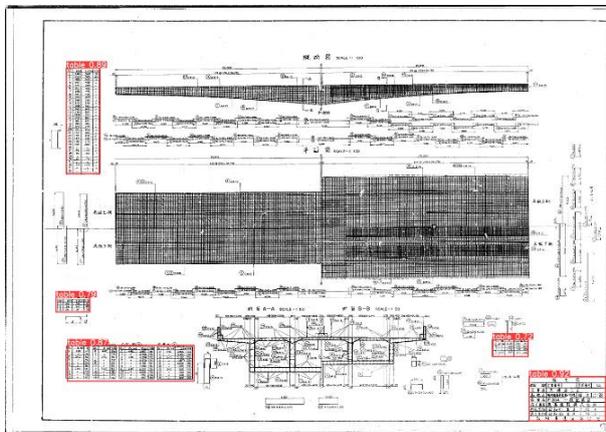


図6 100枚での推論結果

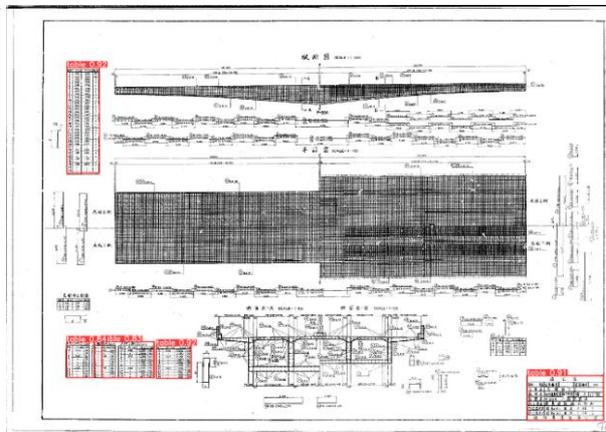


図7 300枚での推論結果

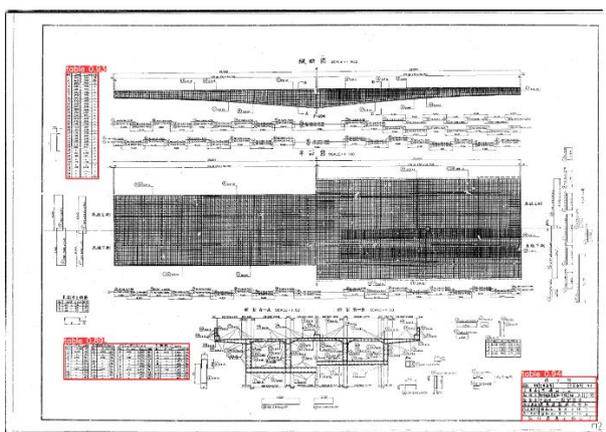


図8 1000枚での推論結果

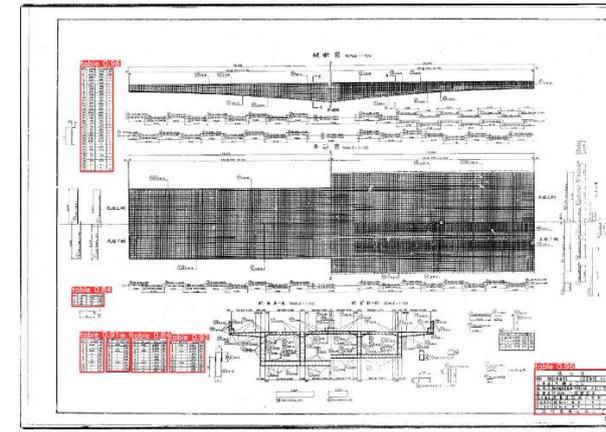


図9 6000枚での推論結果

図4は epoch 数ごとの損失関数(loss)の推移, 図5は epoch 数ごとの precision の推移, 図6～図9は推論結果を表している. これらの結果から, データ数が多いほど良い推論結果が出ている. 6000枚のケースでは precision が 0.98 を超えており, 誤検知は 2%ほどとなり, 十分な結果が出ているといえる. この検知画像では, 全体として大きな差はなかったが, 左下の表に違い

があった。これらの図は4つの表が横並びになっているが、100枚と1000枚のケースでは一つの表として検知されており、300枚のケースでは3つの表として推論されている。6000枚のケースでは正確に推論されていた。これらからデータ数は多いほうが精度は高くなることが分かった。しかし、学習に使う教師データ数を増やすためには、大量の教師データを作成するために、かなりの時間と労力が必要になる。さらに、当然のことではあるが、データ数が多いほど学習にかかる時間が長くなる。実際に本研究でも、データ数が100枚のケースでは21分、6000枚のケースでは22時間と、学習時間には大きな違いがあった。データ数が少なければ精度が落ち、データ数が多ければ時間とコストがかかるが、一度重みデータを作ってしまうと、ほかの設計図に対して検出できるので、多めに設定するほうが望ましいと考える。

次にYOLOの4つのモデル(s, m, l, xモデル)の精度を比較する。学習条件は学習回数(epoch数)300回、データ数は1000枚とする。

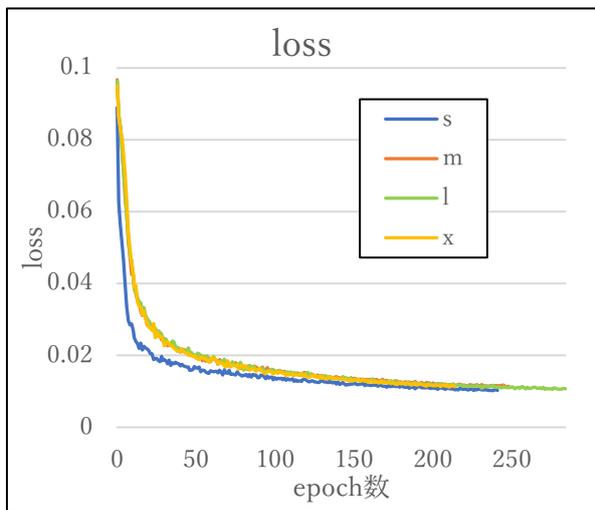


図10 モデルによる loss の比較

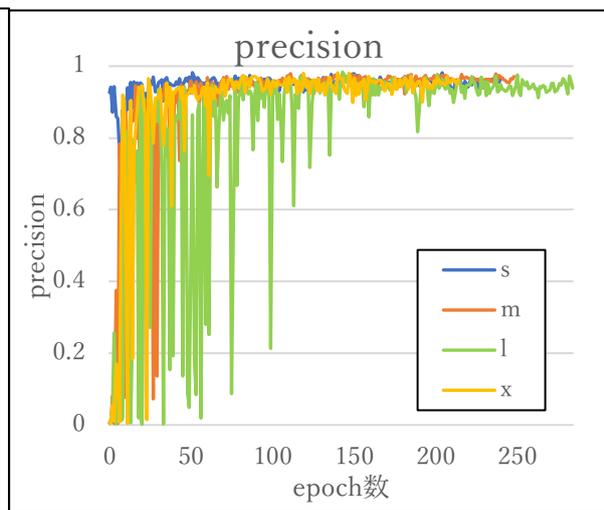


図11 モデルによる precision の比較

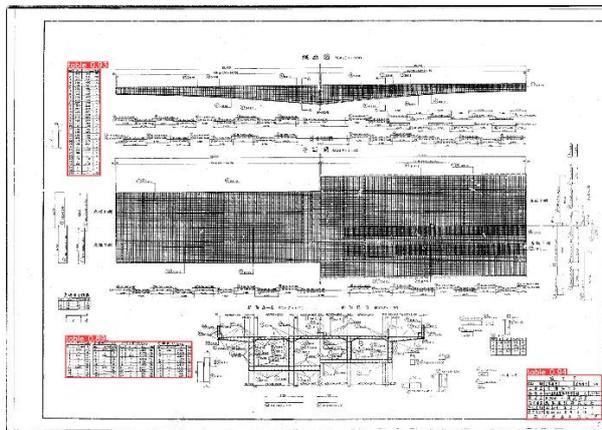


図12 sモデルでの推論結果

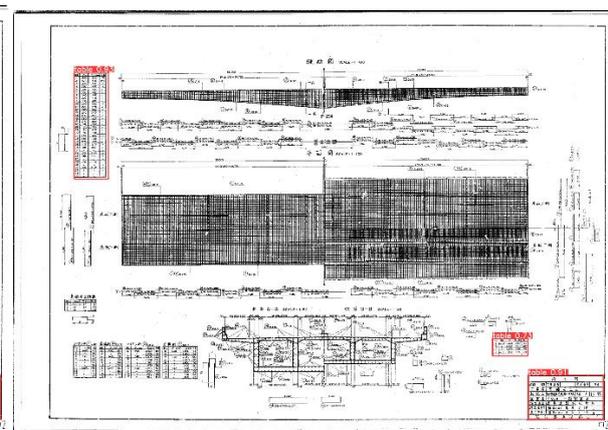


図13 mモデルでの推論結果

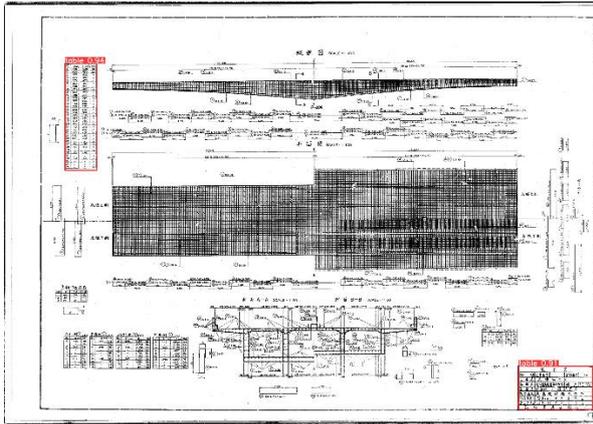


図 14 1モデルでの推論結果

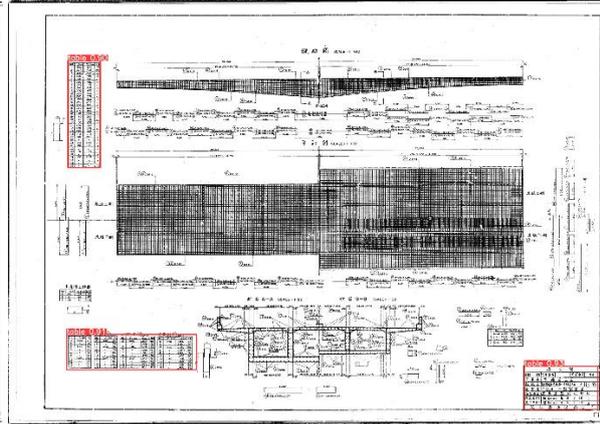


図 15 xモデルでの推論結果

損失関数(図 10), precision(図 11), のすべてで同程度の値をとり, 結果に違いはなかった. 検知画像では s モデル(図 12)と x モデル(図 15)では同じような結果になり, m モデル(図 13)と 1 モデル(図 14)でも似たような結果となった. 学習にかかった時間は, 過学習防止のための自動停止機能(Early Stopping)を使っているため直接比較することはできない. 仮に Early Stopping を使わず 300 回すべて学習を進めた場合, 最も軽い s モデルで 4.06 時間, 最も重い x モデルで 4.53 時間であったと推測でき, 実質的に 30 分の時間差が生じていた. 最も精度の良い結果となった 6000 枚をそれぞれのモデルで学習させた場合, 推定ではあるが, s モデルと x モデルで 3 時間の差が生じる可能性がある. しかし, その時間コストと精度を比べた時, x モデルにすることによる優位性は感じられない. したがってモデルごとの比較では s モデルで学習を進めることが望ましい.

5. まとめ

本研究では都市丸ごとシミュレーションのためのデジタルツイン作成を目的として, ディープラーニングを用いた物体検知アルゴリズム YOLOv5 を用いて設計図面内の表の抽出を行った. その結果, かなり高い精度で表の抽出を行うことに成功した. そして, 様々な条件下で学習させることによって, 最適な学習方法を調査した. 結論として YOLOv5 を用いた最適な学習の方法はデータ数は数千枚規模とし, 学習回数を示す epoch 数は 300 回, 学習に使うモデルは s モデルを使用することが望ましいことがわかった. また検出に使う閾値は誤検出されていた部分の確信度は 0.8 を下回っていることから閾値 0.8 で検出した場合, 誤検出箇所が消去できると考える.

本研究の成果をうけて, デジタルツイン作成を目指して研究を次の段階に進めていく. 目下の課題の一つは, 表内の文字認識である. 今回検知した表の中には重要な数字・文字情報が含まれているため, ディープラーニングあるいは文字認識ライブラリを使ってテキスト・数値データを取り込む必要がある. 次の課題は, 今回の表には含まれない設計図本体を読み取ることである. 図面には構造物本体の線や補助線・文字が混じっており, これらの線がかすれている箇所もあるため, これらの要素を適切に区別してデータを取り込むことは非常に難しく, 要素技術をさらに開発する必要がある. 我々は引き続きデジタルツイン作成を目指して, 1つ1つの要素技術の開発を積み重ねていく計画である.

謝辞：本研究は、阪神高速道路株式会社との「大規模計算機による道路高架橋の地震時応答解析の高度化に関する共同研究」の支援を得て進められたものである。

参考文献

- 1) Alsmiley, ディープラーニングを用いた画像の物体検出とは？モデルや応用例を紹介
https://aismiley.co.jp/ai_news/detection-ai-and-deep-learning-model/
- 2) Glenn Jocher, ultralytics/yolov5, <https://github.com/ultralytics/yolov5>
- 3) Qiita, 物体検出の評価指標 IoU の計算方法
<https://qiita.com/shoku-pan/items/35eae224c59989957623>
- 4) 榊谷英弥, ディープラーニングによる画像推論技術を用いた図面要素抽出の研究, 卒業研究, 神戸大学市民工学科, 2021
- 5) 出口直輝, ディープラーニングによる物体検知技術 YOLOv5 を用いた図面抽出技術の研究, 卒業研究, 神戸大学市民工学科, 2022

筆者:1) 出口 直輝, 神戸大学工学部市民工学科, 学生;2) 銭谷 誠司, 都市安全研究センター, 特命准教授;3) 名下 宥佑, 神戸大学大学院工学研究科市民工学専攻, 学生;4) 榊谷 英弥, 京都大学大学院工学研究科社会基盤工学専攻, 学生

Table detection in highway completion drawings using an object-detection algorithm YOLOv5

Naoki Deguchi
Seiji Zenitani
Yusuke Nashita
Hideya Masutani

Abstract

The concept of the Urban Digital Twin is to recreate an actual city in a virtual space. By conducting various simulations using the Urban Digital Twin, it will be possible to predict damages to the city by a large-scale disaster, and then to minimize risks and damages by disasters. In order to create a digital twin, it is necessary to import information of the structures from two-dimensional blueprints. To this end, we aim to detect tables from a scanned image of highway drawings provided by Hanshin Expressway Co., Ltd., because the tables contain meta information of the drawings and of the structures. We use an object detection algorithm YOLOv5 to detect tables in the images. We study key factors to maximize the detection accuracy, such as the number of training data and network models.

©2022 Research Center for Urban Safety and Security, Kobe University, All rights reserved.