

スマートフォンを用いた軽量な道路損傷の自動検出

今西 智也[†] 宮森 恒[†]

[†] 京都産業大学 〒603-8555 京都府京都市北区上賀茂本山

E-mail: †{g1744140,miya}@cc.kyoto-su.ac.jp

あらまし 本論文では、スマートフォン上で道路損傷を自動検出する問題に取り組む。道路の老朽化問題や、運転支援技術の安全性などを考慮すると、身近に存在するスマートフォンを利用して運転者や公的機関に道路損傷箇所を即時に通知することは重要である。しかし、スマートフォン上での従来の道路損傷検出では、検出時間が1.5秒程度、検出精度が75%程度と、実際の運転時の通知を考慮すると必ずしも十分な検出速度や検出精度とはいえない。そこで、本稿では、小規模なデータセットから物体検出モデルの転移学習と枝刈りを用いることで、スマートフォン上で軽量に道路損傷検出する手法を提案する。実験の結果、精度を大きく落とすことなく、処理時間を短縮化できることを確認した。

キーワード 道路損傷, 物体検出, 転移学習, 枝刈り, 機械学習

1 はじめに

現在、日本国内における自動車の保有台数は、年々増加しており、日常生活には欠かせないものとなっている。車の移動には欠かせない道路の新規敷設や道路工事は各地で行われている一方で、道路の老朽化もまた、新たな問題となっている。特に、1964年に開催された東京オリンピックや、1970年に開催された大阪万博などの開催に合わせて、緊急的に整備された道路や、沿岸や水中部などの立地条件の厳しい場所などでは、老朽化が著しく見られている[1]。日本の道路の新規建設や修繕舗装には、基礎となる地盤が整備された後、アスファルトが主に使用されている。その他の道路の舗装においては、アスファルト舗装の他にコンクリート舗装が存在する。コンクリート舗装はアスファルト舗装と比べ、維持管理コストが安価となっており、道路の耐久性においても50年以上問題なく利用できる特徴を持つ一方で、建設の際に養成期間を必要とし、長期の道路の通行止めをせざるを得ないことから、短期間の施工時間で済む、初期コストの低いアスファルト舗装が採択されやすかった背景がある。アスファルト舗装道路の損傷はわだち掘れやひび割れなどが主にあげられ、原因として大型車両などの重量のある車両の交通によって路面が痛むことによって、道路損傷が発生し道路の修繕が必要になってしまう。近年では社会の発達に伴い、道路をはじめとする各種インフラの老朽化が進んでおり、維持修繕が急務となっており、定期点検診断の省力化も強く求められるようになっている。

供用中のアスファルト舗装の劣化状態を適切に把握することは、道路の舗装修繕の優先度の決定などにおいて大きく貢献でき、より効果的な維持管理が可能となる。道路の老朽化には路面の穴ぼこや陥没、ひび割れなど様々な種類があり、これらは車の乗り心地を悪くするだけでなく、車両が高速走行中にタイヤが損傷箇所にはまってしまった場合、重大な事故に発展する場合もあり道路維持は日本のインフラを支える上で重要である。

しかし、一般に道路の損傷の発見は日本道路協会の道路の定期点検によって実施されており、一般人が道路損傷を発見し報告するためには、道路損傷の修繕は専用の電話番号への通報や各市町村ごとに受け付けている道路損傷発見の専用の受付窓口への報告など、手間が多く、その報告数もわずかである。したがって、道路の利用数の割に効率的に道路損傷の報告ができていないのが現状である。さらに、日本道路協会が現在実施している道路の定期点検についても、目視によって点検が行われており、日本全国の道路の隅々を点検することは不可能である。

道路損傷検出についてはこれまで研究されており、カメラを使った検出の他、特別な機材を使用した研究[2][3]がされてきた。しかし、これらは機材の購入やその維持管理に費用がかかってしまう点において、導入が困難であった。そこで、スマートフォンの普及率が8割を超え、他者への連絡だけでなくナビゲーションや携帯ゲームとしても利用されている日本において、スマートフォンを使用した道路損傷の検出を実現させることによって、より安易に、より素早く道路損傷検出が可能になると考えられる。

そこで本論文では、スマートフォン上で動作する軽量な道路損傷検出に取り組む。これにより、特殊な機材やPCなどの計算能力が高い機材を必要とせず、低コストなリアルタイム道路損傷検出が可能になり、一般人や道路保全者による効率的な道路損傷情報の収集が可能になることが期待される。加えて、リアルタイムでの道路損傷の検出は、自動運転車に搭載された場合には直前で回避行動を取るなどの、自動運転車両の運転性能の向上につながるだけでなく、自動車や自動車に乗っている人へのダメージを減らすことができると考える。

本論文では、転移学習と枝刈りを用いることで、スマートフォン上で軽量に道路損傷検出する手法を提案する。まず、従来研究で作成された道路損傷データセット[4]に対して、データ数が特に少ない特定クラスが含まれる画像をデータ拡張で5倍に増強し、このデータセットを用いて、事前学習済みの物体検出モデルをファインチューニングした。得られたモデルに対

して、モデル圧縮手法の1つである枝刈りを実施し、モデルの軽量化を図った。実験では、枝刈りの割合について複数の割合を比較検討することによって、枝刈りの割合による精度や検出処理速度への影響についても調査した。その結果、従来研究のスマートフォン上での道路損傷検出と比較して、検出時間を大きく短縮することができた。

本論文の貢献は以下のとおりである。

- 本論文では、スマートフォン上で軽量の計算量で高速に処理できる道路損傷検出モデルを提案した。
- データ拡張と枝刈りを併用することで、検出精度を大きく落とすことなく、処理時間を短縮化できることを確認した。

本論文の構成は次のとおりである。第2節では関連研究として、道路損傷を検出するための研究について、特別な機材を用いて路面状況を把握する研究や、スマートフォンを用いた道路損傷の検出の従来研究について説明する。第3節では道路損傷検出の問題設定と使用する物体検出モデルと検出精度や検出速度にアプローチするための転移学習と枝刈りについて説明する。第4節では機械学習の際に使用した学習データセットについて説明する。第5節では第4節で説明したデータセットを用いた評価実験を通して提案手法の有用性を確認する。第6節では本論文の結論とともに今後の課題について説明する。

2 関連研究

本節では、道路損傷検知に関連する研究として、2.1節で特殊な機材を用いた研究を紹介し、2.2節では機械・深層学習による道路損傷検出の従来研究を紹介する。

2.1 特殊な機材を用いた道路損傷検出

特殊な機材を用いる研究としては、光の反射の仕方によって、路面の状況を判定する研究[2]が存在する。これは車両に搭載した発光装置によって路面に向かって光を照射し、その光の反射の仕方によって光センサが路面が損傷している場所を識別している。また、高出力レーザー光プロジェクターとカメラのハードウェアを使った路面状況を把握する研究[5]が行われている。この研究内で提案されたLCMSシステムが多様な道路のデータを取得し、同システムが道路損傷の自動検出と自動分類を行っている。これらの研究は画像データを用いておらず、本論文の道路損傷箇所の検出においては、スマートフォン上で撮影した画像データを用いているという点において本論文とは異なる。

2.2 機械・深層学習による道路損傷検出

機械・深層学習の技術の進歩により、深層畳み込みニューラルネットワークは道路損傷検出の主流技術となっている。代表的な従来研究を紹介する。

スマートフォンを用いた検出として、加速度センサを用いた研究[6]が存在する。この研究では、スマートフォンの加速度センサを利用した路面状況の把握をしているが、カメラ画像での道路損傷検出ではない点において、本論文と異なる。前田氏ら[4]は道路損傷データセットである、Road Damage Dataset 2018を提案した（以下、RDD 2018と記載する）。さらにそ

のデータセットを使用し、MobileNetなどの物体検出モデルで学習させ、学習済みモデルをスマートフォンに実装し、スマートフォン上でリアルタイムで道路損傷を検出できるアプリを制作した。その検出性能は最も良い結果の出たクラスで、PrecisionとRecallが0.75を超え、検出速度は1.5秒であった。本論文のデータの拡張を施す前の学習データセットは前田氏らによって提案されたRDD 2018を使用しており、同じである一方で、本論文の物体検出モデルはYOLOv5を使用して転移学習によってモデルを構築しており、物体検出モデルや学習方法が本論文とは異なる。前述の前田氏らによって提案されたRDD 2018を使用している研究として、Wenzhe Wangらによって発表された研究[7]を紹介する。この研究ではバックボーンネットワークとして、2つのFaster R-CNN(ResNet-152とResNet-101)を比較している。実験についてはGPUを搭載したPC上で処理され、その結果はバックボーンネットワークとしてResNet-152を使用したFaster R-CNNの方がResNet-101を使用したFaster R-CNNよりも優れていることが確認され、これはより深いバックボーンネットワークが抽象的な表現を学習できるためであると考えられた。一方で、計算リソースの限られているスマートフォン上での検出処理を行っているという点において本論文とは異なる。

データ拡張を行った道路損傷検出の代表的な従来研究について紹介する。道路損傷データセットの特定のクラスのデータ数が少ないため、それを補うことを目的としたデータ拡張はこれまでも従来研究として取り組まれており、その代表的な例を紹介する。GANを用いて、実際の道路損傷画像から疑似道路損傷画像を生成し、ポアソンブレンドによって損傷のない道路画像に画像を貼り付け、データセットを拡張する従来研究[8]がある。データ拡張については、本論文でも用いており、データ数が特に少ないクラスを補っている一方で、本論文の物体検出モデルはYOLOv5を使用して転移学習によってモデルを構築しており、物体検出モデルや学習方法が本論文とは異なる。

3 提案手法

3.1 問題設定

本稿で取り組む道路損傷検出の問題を説明する。本問題は、入力された路面画像に対して、道路損傷箇所のバウンディングボックスとその損傷クラスの組の集合を、スマートフォン上でリアルタイム出力する問題である。道路損傷のクラスは、前田氏らによって構築されたRDD 2018[4]に準拠しており、8種類存在する。損傷クラスはそれぞれ、D00:縦方向のひび割れ、D01:道路工事による縦方向の境界線、D10:横方向のひび割れ、D11:道路工事による横方向の境界線、D20:亀甲状のひび割れ、D40:ポットホール、D43:横断歩道のかすれ、D44:白線のかすれである。図1に各クラスの道路損傷の参考画像を示す。

また、スマートフォン上でのリアルタイム検出処理における問題を説明する。本問題では、車が走行している状態でのリアルタイム検出をする必要がある。本論文内での車の道路損傷検



図 1 道路損傷データセットの 8 クラス

出が可能な走行時速の定義は、車が一般道を走行していることを前提とし速度 40km/h とした。また、その際に必要となる検出処理速度は、車が 40km/h で走行している際に 1 秒間で進む距離は約 11 メートルであり、車の前方 2 メートルを道路損傷の漏れがなく検出するためには、5FPS 以上 (画像 1 枚あたり 200ms 以下) 必要である。したがって、本論文の道路損傷検出における処理速度は 5FPS (画像 1 枚あたり 200ms) を目指す。

3.2 基礎となる物体検出モデル

本節では、提案手法の基礎となる物体検出モデルについて説明する。

a) YOLO の概略

まず、YOLO シリーズについて説明する。YOLO(You Only Look Once) とは、深層学習に基づいた物体検出モデルである。これまで、物体検出手法として用いられてきたモデルは、物体領域を推論し、その後にクラス分類の推論を行うため、物体検出に時間がかかってしまうことが課題であり、物体検出について検出処理速度と検出性能は常にトレードオフの関係性にある。しかし、YOLO は入力された画像をあらかじめ指定されている正方形にリサイズした後、入力画像をさらに細かく、グリッドと呼ばれる正方形に分割することで、リアルタイム物体検出を可能としており、最初に提案された YOLOv1 [9] から 2022 年 7 月に発表された YOLOv7 [10] の 2023 年 1 月現在に至るまで開発が続けられている。また、バウンディングボックス

(Bounding Box) と呼ばれる正解情報を含めた矩形の中心座標、幅、高さの情報から重みデータを作成し、得られたデータの特徴量から矩形を抽出している。

b) YOLO の調査

前節でも記述した通り、YOLO は現在に至るまで様々なバージョンが発表されており、その性能も古いものよりも改善されているが、スマートフォン上での検出処理性能は最も新しいものが必ずしも優れているものであるとは限らない。そこで、基礎となる物体検出モデルを選択するにあたって、リアルタイム検出処理を可能とするために検出処理時間と、スマートフォンの限られたストレージを節約するために学習済みモデルのサイズに着目して調査を行った。なお、検出処理時間については、計算能力の高い PC 上の処理ではなく、リソースが限られているスマートフォン上での計算時間を参考にした。また、学習済みモデルのサイズについては、学習済みモデルサイズと、その学習済みモデルをスマートフォン上で実装するための変換処理した後のモデルサイズを比較した。比較した物体検出モデルについては、YOLO シリーズの中でも特にリアルタイム処理を得意としている、YOLOv5 と YOLOv7 [10] について比較調査を実施した。YOLOv5 は 2020 年に公開されたモデルであり、ネットワークの大きさに応じた n, s, m, l, x の 5 種類のモデルが用意されている。最も小さい n モデルから最も大きい x モデルになるにつれて精度が高くなり、学習時間と検出時の処理時間が増える傾向にある。一方で、最も小さい n モデルでは上位のモデルと比較すると精度が劣るが、検出時の処理時間は短縮され、リアルタイム処理に向いていると言える。YOLOv7 は 2023 年 1 月現在で最も新しい YOLO のモデルであり、YOLOv5 と同様に精度や推論時間を考慮した複数種類のモデルが用意されている。以上を踏まえて、YOLOv5 から n と s モデルを選択し、YOLOv7 からはモデルの中で最も軽量である tiny モデルを選択して調査を行った。表 1 に調査の結果を示す。

表 1 YOLOv7 と YOLOv5 の比較

モデル名	モデルサイズ (変換前)	モデルサイズ (変換後)	1 枚あたりの 処理速度
v7-tiny	12.1MB	24.9MB	675.3ms
v5s	14.4MB	28.6MB	895.45ms
v5n	3.8MB	7.3MB	397.3ms

調査の結果、モデルサイズについては、スマートフォン実装のためのモデル変換前とモデル変換後のいずれについても YOLOv5 n モデルが最小であった。これは限られたスマートフォンのストレージを節約することに寄与すると考えた。また、学習済みモデルのスマートフォン上での検出処理時間についても、3 種類のモデルの中では YOLOv5 n モデルが最も短いことが示された。これは計算量が少ないことを示しており、リア

ルタイム処理に向いている物体検出モデルであると考えられる。したがって、本論文では物体検出モデルとして、YOLOv5 n モデルを採用することとした。ここで、図 2 にスマートフォン上の道路損傷検出処理の流れを示す。

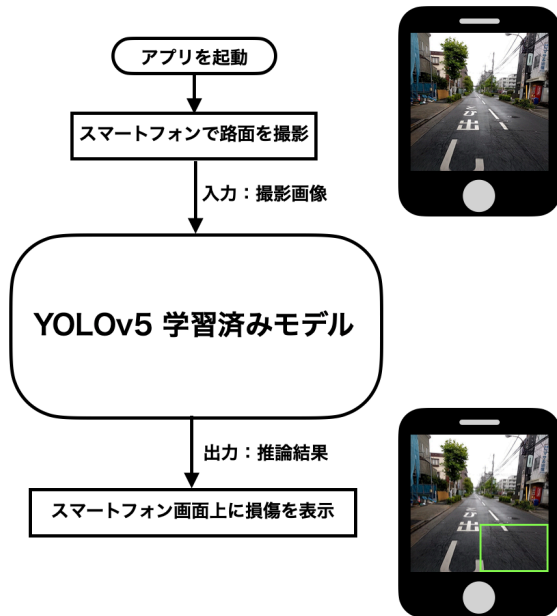


図 2 スマートフォン上の検出処理の流れ

3.3 道路損傷に特化した検出と軽量化

3.3.1 転移学習

本論文で用いる転移学習は、大規模データセットで学習された事前学習済みモデルのパラメータを初期値として、別の目標タスクの少量データセットでこれを再学習することで、短い学習時間で比較的性能の高いモデルを構築する手法である。本論文では、MS COCO データセット [11] で事前学習された YOLOv5 n モデルを用いて再学習する。

3.3.2 モデルの圧縮

これまで、スマートフォンなどの限られた計算能力のデバイス上で物体検出などの重い負荷の演算を動作させることは、演算時間が長くなってしまったりメモリ容量が圧迫されることなどが課題であった。このため、モデルの圧縮手法が注目されており、その手法はいくつか存在する。

代表的なモデルの圧縮手法の 1 つである、モデルの量子化手法は、学習済みネットワーク上の重みを、現在使用されているビット数よりもさらに少ないビット数で表現することによってメモリの使用量を低減する手法である。この手法では、学習済みモデルのサイズを小さくすることができる一方で、精度劣化の懸念がある [12] こと、および、本論文の主眼である検出処理時間短縮に直接寄与することができない。

また、別のモデル圧縮手法として、知識蒸留がある。これは、目標タスクにおいて教師となり得る高い性能の大きなモデルが存在するときに、その挙動を真似るように小規模モデルを訓練することで、教師の性能を保持しながら小さいモデルを獲得す

る手法である。ただ、2023 年 1 月現在で道路損傷検出における教師モデルに相当する適切な大規模モデルが存在しなかったため、本論文ではこのアプローチは採用しないこととした。

最後に、モデルの枝刈り (pruning) について述べる。枝刈りは、構築されたニューラルネットワークの一部のニューロンや重みを削除することで、軽量のモデルを獲得する手法である。枝刈りは、削除する構成要素の単位により、unstructured pruning [13] と structured pruning [14] に大まかに分類できる。unstructured pruning は、重み単位で不規則に枝刈りを行う手法であり、structured pruning は、層やフィルタ単位での規則性を持った形で枝刈りを行う手法である。一般に、unstructured pruning では、絶対値が小さい重みを 0 に変換することで枝刈りをするため、計算量は変化せず、演算処理の効率化にそのままの形で寄与するのは困難である。structured pruning では、演算処理の効率性は高いものの、unstructured pruning と比較して精度が低くなる傾向がある。以上を踏まえ、本論文では、structured pruning である Neuron Merging [14] を採用し、学習の過程で枝刈りされたモデルを Fine-Tuning によって再学習することとした。Neuron Merging は、枝刈りされた CNN フィルタによる本来の次元の出力を、枝刈りされていないフィルタによる次元の出力に併合することで、枝刈りされたフィルタの影響を補償する手法である。これにより、structured pruning における精度の劣化を軽減することができると期待できる。

4 データセット

本論文では、RDD 2018 [4] データセットを用いる。本データセットには、8 種類の道路損傷クラスが付与された 7240 枚の訓練用道路画像が含まれている。しかし、一部クラスのデータ数が十分ではないため、データの偏りによる過学習が懸念される。本論文では、データ数不足を補い、過学習を軽減する方法としてデータ拡張を実施した。具体的には、特にデータ数が少ない「D10」、「D11」、「D40」、「D43」のラベルが付与された訓練画像集合を 5 倍に増加させた。増加させる際は、「縦方向へのシフト」、「横方向へのシフト」、「画像の反転」、「明度の調整」を適用した。データ拡張前とデータ拡張後の各クラスの訓練画像数を図 3 に示す。

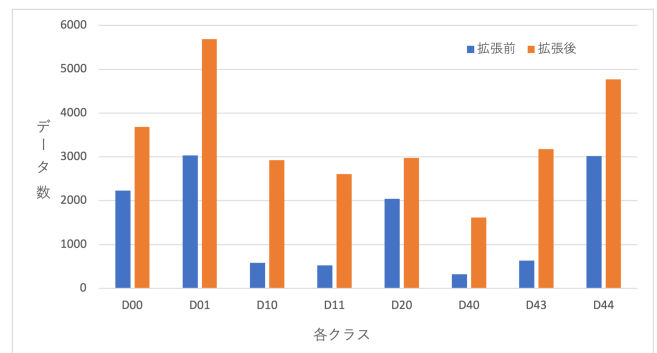


図 3 データ拡張によるデータセット比較

5 実験と評価

本節では、提案手法と従来手法の性能を比較し評価する。まずは、検証すべき研究課題を列挙した後、実験設定について説明し、最後に実験結果を示し考察する。

5.1 研究課題

以下の研究課題について検証する。

RQ1 物体検出モデルの転移学習によって、道路損傷をどの程度正確に検出できるか？

RQ2 物体検出モデルへの枝刈りによって、リアルタイム道路損傷検出の精度と処理速度はどのように変化するか？

RQ3 データ拡張を行うことで道路損傷検出の性能は改善されるか？

5.2 実験設定

本節では実験設定として、評価指標と従来研究の手法、使用したスマートフォンについて説明する。

a) 評価指標

学習後のモデルは、Precision, Recall, mAP, モデルサイズ, 1枚あたりの処理速度を指標として評価する。なお、本論文ではスマートフォン上でのリアルタイム道路損傷検出処理を目標に設定しているため、モデルサイズについてはスマートフォン上で実装するための変換処理した後のモデルサイズを参考にし、処理速度の評価指標については、PC上での動作ではなく、スマートフォンに搭載した際の画像1枚あたりの検出処理時間を参考にすることとした。

b) 比較手法

本実験で比較するスマートフォンを用いた道路損傷検出の従来手法について以下で説明する。従来手法と提案手法の諸元を表2に示す。

スマートフォンを用いた道路損傷検出の研究 [4] この従来研究は道路損傷の検出をスマートフォン上で行っている。物体検出モデルは複数の物体検出モデルを使用し比較しており、SSD MobileNet が最も良い性能を示した。使用しているデータセットは、本論文が元にした RDD 2018 データセットであるが、本論文のデータ拡張は行われていない。また、学習済みモデルへの入力画像は、元々のデータセットの画像解像度 600×600 ピクセルを、 300×300 ピクセルにリサイズした画像を用いている。

表 2 本実験で比較する手法

物体検出モデル	データセット	入力画像サイズ
従来研究 [4] SSD MobileNet	RDD 2018	300×300
提案手法 YOLO v5 n	RDD 2018 + データ拡張	600×600

c) 実行環境

実行環境について説明する。スマートフォンとしては、Google

Pixel 4a を用いて処理速度等を計測した。図 4 にスマートフォン上で道路損傷を検出している様子を示す。



図 4 スマートフォンによる、道路損傷検出の様子

5.3 実験結果

表 3 に、枝刈りの割合変化に伴う検出精度、モデルサイズ、検出速度の結果を示す。表 4 に、最も高速に動作した、枝刈りの割合を 0.2 にした際のモデルの性能と従来研究との比較を示す。また、正しい検出例、誤検出の例、検出漏れの例をそれぞれ図 5 に示す。

表 3 異なる枝刈り割合に伴う提案手法の性能の変化

枝刈りの割合	Precision	Recall	mAP	モデルサイズ	処理速度
0.0	0.593	0.560	0.582	7.3MB	397.3ms
0.1	0.579	0.525	0.546	5.5MB	366.5ms
0.2	0.495	0.562	0.540	5.9MB	359.0ms
0.3	0.498	0.554	0.533	5.4MB	361.6ms

表 4 従来手法と提案手法の比較

	Precision	Recall	入力画像サイズ	処理速度
従来手法	0.811	0.475	300×300	1500.0ms
提案手法	0.495	0.562	600×600	359.0ms

5.4 考察

実験結果より、学習済みモデルに枝刈りを施すことで、モデルのサイズを軽量化しスマートフォンの限られたストレージを節約することに加えて、その検出処理速度においては枝刈りをしないモデルと比較して最大で 38.3ms、従来研究と比較して 1141ms 短縮できることを確認した。一方で、検出精度については、データ拡張により多少の性能向上は見られたが、性能が

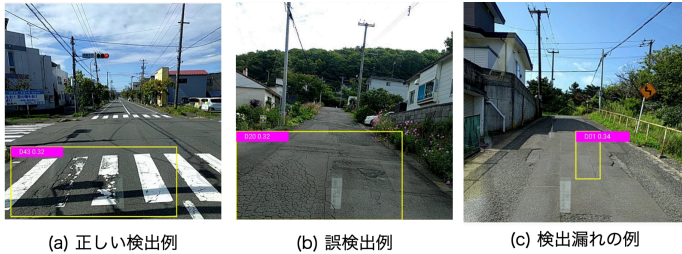


図 5 提案手法による検出例

大きく向上することはなかった。これは、データ拡張において「縦方向へのシフト」、「横方向へのシフト」、「画像の反転」、「明度の調整」の操作しかしておらず、元画像と比較して、あまり大きな変化がなかったことが原因の一つであると考えられる。また、図 5 (a) に示す横断歩道のかすれのクラスはコントラストが明確な場合が多く、正しい検出例が多く確認できた。誤検出の例である、図 5 (b) では地面のくぼみの道路損傷を別の損傷と誤って検出する例が多く見られた。検出漏れの例である、図 5 (c) では地面のくぼみの道路損傷の検出を漏らしてしまう例が見られ、くぼみの輪郭が必ずしも明瞭ではない場合が多く、これが一因と考えられた。

6 ま と め

本論文では、スマートフォン上で動作するリアルタイム道路損傷検出のため、モデルの圧縮手法である枝刈りとデータ拡張を用いて、軽量の演算とモデルサイズで学習済みモデルを構築する手法を提案した。実験結果より、枝刈りによって、検出精度を過度に落とすことなく検出速度を向上させた、より軽量の検出モデルを獲得できることを確認した。今後は、より高速で検出漏れの少ないモデルへの改善、今回想定した速度よりも高速で走行する車両を想定したモデルへの改善が課題である。

謝 辞

本研究の一部は科研費 18K11557 の助成を受けたものである。ここに記して感謝の意を表します。

文 献

- [1] 道路の老朽化対策の本格実施に関する提言. 社会資本整備審議会道路分科会建議, 2014.
- [2] Christoph Mertz. CONTINUOUS ROAD DAMAGE DETECTION USING REGULAR SERVICE VEHICLES. 2011.
- [3] Rui Fan, Ming Liu. Road Damage Detection Based on Unsupervised Disparity Map Segmentation. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2019.
- [4] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiya, Hiroshi Omata. Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. CoRR, 2018.
- [5] John Laurent, Jean Francois Hebert, Daniel Lefebvre, Yves

Savard. 3D laser road profiling for the automated measurement of road surface conditions and geometry. In 7th RILEM international conference on cracking in pavements 2012.

- [6] 白井 悠, 井原 務, 村上 浩. スマートフォンを用いた舗装の路面評価手法に関する一提案. 2014.
- [7] Wenzhe Wang, Bin Wu, Sixiong Yang, Zhixiang Wang. Road Damage Detection and Classification with Faster R-CNN. In 2018 IEEE international conference on big data (Big data), 2018
- [8] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiya. Generative adversarial network for road damage detection. Computer-Aided Civil and Infrastructure Engineering, 2020.
- [9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. CVPR2016, 2016.
- [10] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv:2207.02696v, 2022
- [11] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar. Microsoft COCO: Common Objects in Context. In European conference on computer vision, 2014.
- [12] 山本 康平, 橘 素子, 前野 蔵人. ディープラーニングの軽量化技術. ディープラーニングの軽量化技術 2019 年 5 月/第 233 号 Vol.86 No.1, 2019.
- [13] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, Hans Peter Graf. Pruning Filters for Efficient ConvNets. In ICLR, 2017.
- [14] Woojeong Kim, Suhyun Kim, Mincheol Park, Geunseok Jeon. Neuron Merging: Compensating for Pruned Neurons. In 34th Conference on Neural Information Processing Systems (NeurIPS 2020), 2020.