

原木材計測への深層学習の応用

○森井隆慎 (北海道大学) 江丸貴紀 小林幸徳 Ankit A. Ravankar

Application of Deep Learning to Raw Woods Measurement

○ Takayoshi Morii*, Takanori Emaru**, Yukinori Kobayashi**, Ankit A. Ravankar**

*School of Engineering, Hokkaido University

**Faculty of Engineering, Hokkaido University

Abstract: There is a significant demand for efficient management of raw wood logs in forestry. At present, the management of raw wood logs is mostly performed manually. To overcome this manual process there is a demand for development of automatic wood log counting system. In this research, we propose a counting and detection system of raw wood logs using computer vision and deep learning algorithms. The data set for learning was obtained from actual forest work site and was corrected for accurate counting. We also propose the software development on tablet PC for ease of use in assisting workers to count the wood logs accurately.

1. はじめに

労働力不足が深刻化する中、林業においても業務の効率化が大きく期待されている。特に原木材を管理する現場においては人手により Fig. 1 のような多数積載された木材の本数を計数しており、トラック上での作業は危険で夏場は非常に過酷である。そこで本研究では多数の積載木材を断面方向から撮影することで画面内の木材を深層学習により自動計数できるようなシステムを開発した。丸太画像計測システム¹のような製品が提案されているが、本研究では認識領域の手作業による指定が不要でタブレット端末上でも自動認識を可能とするために深層学習の適用および学習における工夫を行った。深層学習は Mask R-CNN² を用いて、データセットの作成やパラメータ調整を行った。また、実際の現場ではモバイル端末による本数計数の補助アプリの需要があり、Mask R-CNN を Microsoft Surface Pro で実行できるよう実装した。

2. Mask RCNN による学習と推測

2.1 Mask RCNN のデータセットと学習

原木材の一本一本を区別して計数するためにインスタンスセグメンテーションが可能な Mask R-CNN を用いた。Mask R-CNN を選んだ理由は 2 つある。一つ目は、高速性を必要としないことである。Yolo や SSD などの高速な物体検出手法もあるが、静止画に対する実行で 10 秒程度の認識であれば合理的であると判断したため、Faster R-CNN をベースとした Mask R-CNN を用いた。二つ目は、将来的にそれぞれの直径を求めることができるようにセグメンテーション機能も必要と判断したからである。

データセットについては、現場で取得あるいはインターネットで取得した木材の積載画像 89 枚のうち 12 枚の RGB 画像の学習・推測用画像を選んだ。そのうち 9 枚を学習用とし、反転・輝度変化・ノイズ付加・カットアウト・変形・8 枚の背景画像との合成などにより 2352 セットの RGB 画像とマスク画像を作成した。調整した主要なパラメータは Table 1 にまとめた。プログラムは Mobile_Mask_RCNN³ を参考とし、backbone は mobilenet v1⁴ とした。また、エポック数に対する損失の値を Fig. 2 に示す。



Fig. 1: 積載された原木材

Table 1: Mask RCNN 学習時におけるパラメータ

max image size	640
min image size	640
steps per epochs	200
epochs(heads)	18
epochs(4+)	22
epochs(all)	24
train rois per image	16
rpn anchor scores	(8, 16, 32, 64, 128)
validation steps	6

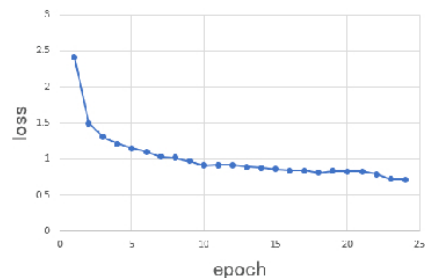


Fig. 2: RGB データでの学習時の損失値の推移

2.2 学習結果の評価

テスト画像として Fig. 3 のように 3 枚の画像を用意し、式 (1) の再現率 R を用いて評価した。また、推測結果



Fig. 3: 左上:テスト画像 1, 右上:テスト画像 2, 左下:テスト画像 3



Fig. 4: テスト画像に対する推測結果

Table 2: テスト画像に対する再現率

テスト画像	R [%]
1	81.8%
2	70.6%
3	80.7%

を Fig. 4 に、テスト画像での精度を Table 2 にまとめた。

$$R = \frac{TP}{TP \cup FN} \quad (1)$$

式 (1) 中の TP は True Positive, FN は False Negative を表す。3 つのテスト画像の再現率の平均は 77.7% である。データセットに使う元の画像を増やすことも必要であるが、mobilenet v1 において epoch あたりの steps を倍にしても再現率に大きな差はなく、データセットやパラメータ調整の他に backbone やアーキテクチャの見直しやマルチモーダルな学習などにより精度向上を検討する必要がある。完全な自動化のためには不十分な精度であるが、現場での計数補助としては十分な精度であるということが調査でわかった。

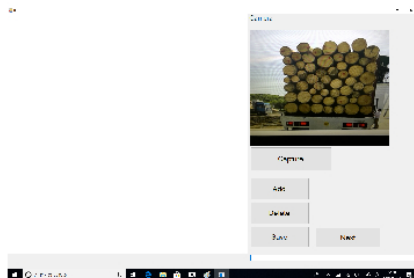


Fig. 5: 実装したアプリの操作パネル



Fig. 6: 実装したアプリの修正操作の様子 左:誤認識部分の修正, 右:修正後の結果画像

3. モバイル端末への実装

モバイル端末による木材の計数補助が現場で求められていることの一つであるため、上述の Mask R-CNN による木材自動計数がモバイル端末 (Microsoft Surface Pro) 上で実行できるかを検証する。アプリの作成には、.NETFramework⁵ を用いて C# により実装した。作成したインターフェイスは Fig. 5 に示す通りであり、システムの誤認識を修正できる機能を Fig. 6 のように実装した。Microsoft Surface Pro(Intel Core i7-7660U, RAM 8GB) 上での推測テストを連続で実行した場合の時間については、1 回目は 11.27 秒、2 回目が 6.51 秒、3 回目が 6.59 秒で自動計測ができており、修正して電子データとして管理できる点や認識精度も 77.7% という点で成果として十分だという評価を得た。

4. 結 論

本研究では多数の木材の自動計数のために深層学習を応用し、モバイル端末への実装をすることで現場への導入の可能性を示した。今後は直径の計測、逆光や複雑な背景などの悪条件下でも高精度に認識できるような手法の検討などが必要である。

謝 辞

本研究は「素材流通の ICT 化に伴う木材検知システム研究」の助成を受けたものである。また、現地調査に際しては株式会社森林環境リアライズ、株式会社ヨシダ、高知おとよ製材株式会社の皆様にご協力いただきました。ここに記して感謝の意を表します。

文 献

- (1) 速測デジ, <http://hayasoku.com/>, (2019 年 9 月 29 日)
- (2) Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, Mask R-CNN, IEEE International Conference on Computer Vision (ICCV), Oct. 2017, DOI: 10.1109/ICCV.2017.322.
- (3) Mobile_Mask_RCNN, https://github.com/-GustavZ/Mobile_Mask_RCNN, (2019 年 9 月 29 日)
- (4) Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
- (5) .NET Framework, <https://docs.microsoft.com/ja-jp/dotnet/framework/>, (2019 年 9 月 29 日)