

# 深層学習を用いた砥粒の検出に関する研究

## Research on Abrasive Grains Detection based on Deep Learning

茨城大 ○西村賢宏 ◎尾島裕隆, 周立波, 清水 淳, 小貫哲平

### 要旨

機械学習を用いて画像に含まれる物体のクラス分類を行う手法の一つが、物体検出である。この手法では、入力画像中の物体の位置の特定とクラス分類を行う。本研究では、砥石の状態による研削特性を分析する手段の一つとして、画像に含まれる砥粒およびその位置の判別を行うことを目指した。今回、深層学習モデルとして EfficientDet を採用し、実際に砥粒の検出を行った結果について報告する。

### 1. 研究背景

研削砥石作業面の砥粒の形状と分布は、工作物の仕上げ面粗さや砥石寿命の判断指数になる他、研削特性にも影響を及ぼす。高精度な研削加工を実現するためには砥石表面を正確に計測することが重要である。一方で、深層学習を利用した物体検出が近年注目を集めている。これまで本研究室では SSD による物体検出を用いた研究を行っていた。本研究は、深層学習を用いた物体検出により砥粒の位置を判別することを目的としている。

### 2. 深層学習による物体検出

深層学習を用いた物体検出は、単一の物体カテゴリを識別するのみの画像認識とは異なり、画像に含まれる複数の物体を同時に識別することができ、位置の特定も行うことが可能である。この手法に用いるモデルは R-CNN(Regional CNN)や SSD(Single Shot MultiBox Detector), YOLO(You Only Look Once)などが例に挙げられる。

SSD は大きく分けて、画像認識モデルの VGG16 をベースとしたバックボーンネットワークと、バックボーンネットワークで得られた情報を基に推論を行う部位の 2 つのブロックに分類することができる。

本研究では、EfficientDet<sup>[1]</sup>と呼ばれる物体検出モデルを使用する。これは画像に含まれる単一の物体のカテゴリ分類のみを行うモデルである EfficientNet をバックボーンに持ち、カテゴリ分類と位置の特定を行うものである。検出を行う全体の流れは図 1 に示すように主に 3 つに分けられる(図 1)。異なる解像度で特徴抽出を行い、デフォルトボックスを生成するバックボーンネットワークのブロックと、抽出した解像度の異なる特徴マップ同士を組み合わせた BiFPN(Bidirectional Feature Pyramid Network)レイヤーのブロック、生成したボックス情報を正解ボックスと比較して学習を行うブロックである。バックボーンネットワークにおけるデフォルトボックスの生成方法は SSD に類似している。まず入力画像に対して各層ごとに解像度によって分割されたグリッドセル全てに対しアスペクト比や大きさの異なる 4 つのデフォルトボックスを生成する(図 2)。デフォルトボックスの大きさは特徴マップの解像度によって異なり、そのスケール  $s_k$  は次の(1)式で定義される。

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1)$$

$$k \in [1, m], s_{min} = 0.2, s_{max} = 0.9 \quad \dots(1)$$

$m$  は特徴マップの総数を、 $k$  は何枚目の特徴マップであるかを指す。

また、1 つのグリッドセルに作成されるデフォルトボックスの内 1 つは他の 3 つのデフォルトボックスとは異なる大きさを持つ。

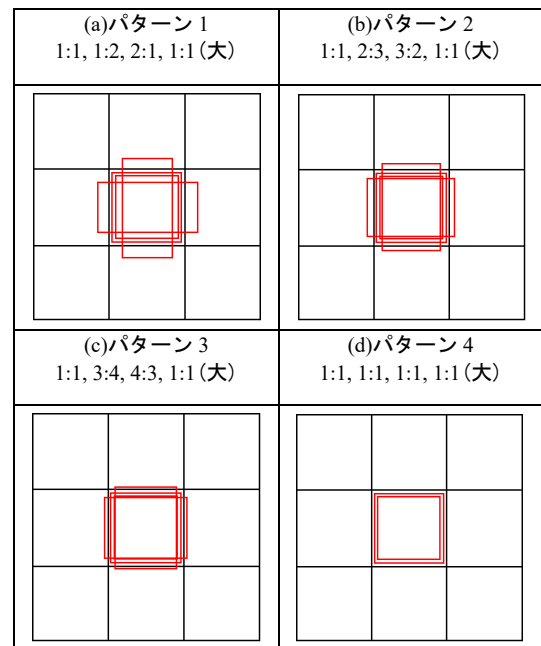


図 2. アスペクト比の組み合わせ

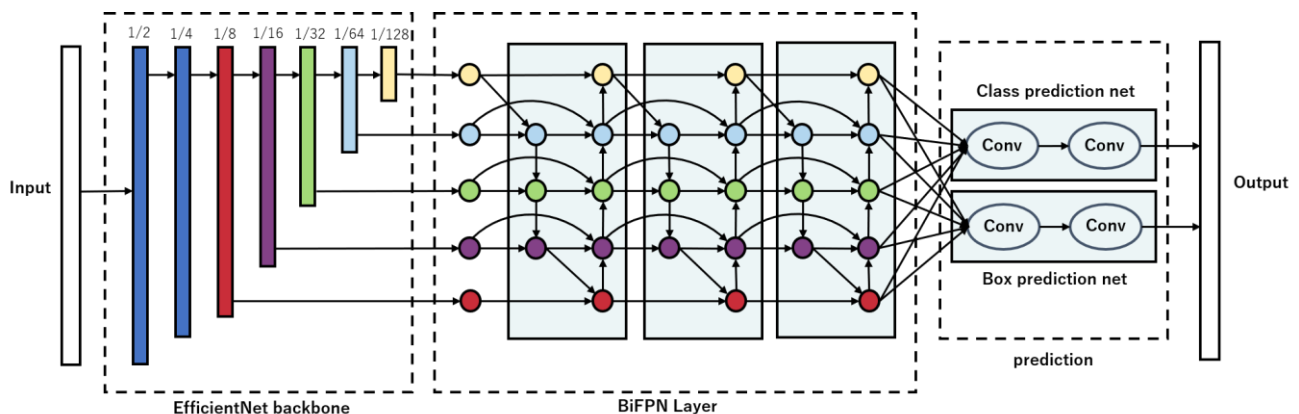


図 1. EfficientDet モデル全体図<sup>[1]</sup>

そのスケール $s_k'$  は次の(2)式で定義される。

$$s_k' = \sqrt{s_k s_{k+1}} \quad \dots(2)$$

各デフォルトボックスの座標とアノテーションデータに含まれる正解ボックスの座標を比較し、デフォルトボックスの形を修正する。修正後のボックスをバウンディングボックスと呼ぶ。出力されたバウンディングボックスの情報と画像から抽出された特徴量はBiFPN レイヤーのブロックで異なる解像度のものと組み合わせられ、最終的に教師データと比較して推論、学習を行う。このモデルで最も特徴的な点は、図1に含まれるBiFPN レイヤーである。BiFPN レイヤーはそれぞれに重み付けを行った解像度の異なる特徴マップ同士を拡大、縮小して組み合わせることで従来の物体検出手法と比較して高速かつ高精度の出力を可能としている。特徴マップを結合する手法は、次の(3)式に表すものを用いる。

$$O = \sum_i \omega_i \cdot I_i \quad \dots(3)$$

$O$ : 出力,  $I$ : 入力,  $\omega$ : 重み

3. 学習条件

学習に使用するデータセットの内訳は訓練用 192 枚、検証用 48 枚の計 240 枚である。詳細を表2に示す。デフォルトボックスのアスペクト比は図2に示すように1:1のスケール $s_k$ と $s_k'$ のものが1つつつを固定とし、他のスケール $s_k$ の2つのボックスのアスペクト比を変更したものを4パターン用意する。尚、EfficientDetのデフォルトのアスペクト比の組み合わせは(d)である。それぞれの学習結果を比較し、砥粒の検出に最適な組み合わせを探す。モデルの評価には砥粒の画像を使用した検出のテストと、Average Precision(平均適合率, AP)及び Average Recall(平均再現率, AR)の値を用いる。AP, ARの定義は次の式4及び表1で表される。

$$Precision = \frac{TP}{TP + FP} \quad \dots(4)$$

$$Recall = \frac{TP}{TP + FN}$$

TPは正解データと予測結果が共に砥粒を検出したことを示す True Positive, FPは正解データ上で砥粒でない箇所に砥粒であると予測された False Positive である。同様にして FN(False Negative), TN(True Negative)が定義される。Average Precision 及び Average Recall は正解データと予測結果が一致した True の場合のみの Precision, Recall の平均値である。

4. 学習結果

4つの条件でそれぞれ学習したモデルを使用した推論により、画像から砥粒を検出するテストを行った。テスト結果を図4に示す。この結果からはパターン2とパターン3が砥粒の検出数及び誤検出の少なさで優れていると考えられる。しかし、表3に示す各モデルの平均適合率(Average Precision)及び平均再現率(Average Recall)で比較した場合、パターン3のアスペクト比の組み合わせが最も優れていると言える。

5. 結言

EfficientDet モデルを用いた学習において、砥粒を検出するのに最も適したデフォルトボックスのアスペクト比を決定した。デフォルトボックスの形状を検出対象の形状に近づけることで、モデルの精度が向上する結果となった。

- 1) Mingxing Tan, Ruoming Pang, Quoc V. Le (2019) "EfficientDet: Scalable and Efficient Object Detection", pp5

表 1. TP, FP, FN, TN

|             | 正解データ Positive | 正解データ Negative |
|-------------|----------------|----------------|
| 予測 Positive | TP             | FP             |
| 予測 Negative | FN             | TN             |

表 2. 学習条件

| params                | value    |
|-----------------------|----------|
| Training : Validation | 192 : 48 |
| Batch Size            | 12       |
| Epoch                 | 1000     |
| Activation Function   | Swish    |

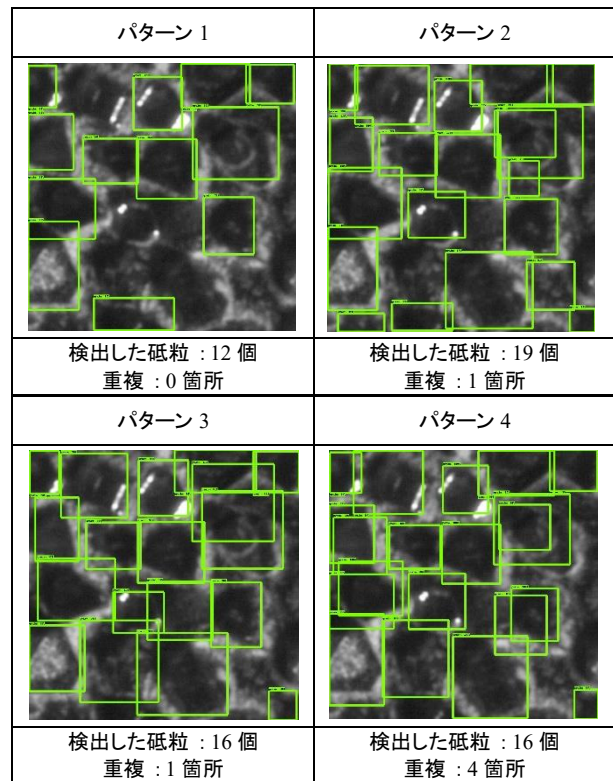


図 4. テスト結果

表 3. Average Precision, Average Recall 比較

| アスペクト比    | Average Precision | Average Recall |
|-----------|-------------------|----------------|
| (a)パターン 1 | 0.514             | 0.695          |
| (b)パターン 2 | 0.567             | 0.680          |
| (c)パターン 3 | 0.575             | 0.683          |
| (d)パターン 4 | 0.515             | 0.639          |