長岡技術科学大学大学院
工学研究科修士論文

題目

Mask R-CNN を用いた自動車ホイールの鋳巣検出に関する研究

指導教員
准教授　杉田　泰則

電気電子情報工学専攻
NGUYEN THIEN ANH
16314682

令和２年２月７日

Master thesis
A study on detecting cavity defects inside automotive wheel using Mask R-CNN

Author: 16314682 Thien Anh Nguyen
Supervisor: Associate Professor Yasunori Sugita

Recently, the automation of the production process in factories is being promoted strongly in order to improve the quality of products, increase the efficiency of the production process and also improve the safety of factory workers. In the automotive industry, casting methods are generally used to manufacture wheels from an alloy of aluminum or magnesium because of the possibility of making complex shapes that would be otherwise difficult or uneconomical to make by other methods.

However, one of the disadvantages of the casting method is the casting product may not be completely solid and there are many cavity defects that exist inside of it. Depending on the amount and location, these cavity defects will be able to affect the safety quality of automotive wheels, which is very important because of the fast running automobile. Because of that, manufactured wheels require internal inspection before selling to customers.

The most used internal inspection method is a visual examination using wheel X-ray images. The X-ray imaging process gives a grayscale image of the product. In that grayscale image, white pixels represent the thickness of the wheel. Thus, where the color suddenly goes darker than its surroundings has the possibility of being a cavity defect. The inspector will detect the cavity defects base on that definition. However, inspection is required for a large number of automobile wheels, and the results differ depending on the skills of inspectors. For these reasons, automation of automobile wheel internal inspection became very necessary. Despite the conduction of many kinds of research, visual examination stills remain as the solution of this internal inspection because of the ambiguous definition of cavity defect. Therefore, high-efficiency technology is very needed.

In the past few years, deep learning technologies have been attracted attention because of its potential for efficiency. This study considered applying a deep learning model called Mask R-CNN, which is one of the instance segmentation methods and can detect objects at pixel-level, to the cavity defect detection inside the wheels, in order to automate the product internal inspection process. As first, we divided the wheel X-ray image, which has a very large size, to smaller images by applying a small sliding window on it. We also set the stride of this sliding window as its half size to increase defect-contain-data and prevent miss when inference. Then, using human-checked mark data as ground-truth, we conducted two experiments.

The first experiment is aimed to confirm that Mask R-CNN can detect cavity defects from wheel X-ray images. Although our target is cavity defects only, these defects exist inside of the wheel. Accordingly, using one wheel type, we compared two models that were trained from data with defect-only labels and data with defect-and-wheel labels. As a result, with test data set of the same wheel type, we confirmed that either model can detect cavity defects from wheel X-ray image as human-level, and the model that trained from data with defect-and-wheel labels is better because of the reduction of overdetection. However, because of training on just one wheel type, the results on other wheel types are not good enough. Therefore, we knew that the detection efficiency is depending on wheel type and proposed to create a dedicated model for each wheel type.

However, labeling data for ground-truth is required time and human resources so there is a trouble of lack of data. For that reason, the second experiment is conducted as a solution for the lack of wheel data. We tried to fine-tune from a model that trained for one wheel type to a model for a different wheel type with less training data. As a result, we found that the number of data required learning can be reduced to below one-sixth, and detection efficiency equal to or higher than that of the dedicated model can be achieved.

目次

図目次

表目次

第 1 章

はじめに

1.1　研究背景

一般に自動車ホイールは，アルミなどを鋳造して製造される．鋳造方式の欠点の 1 つとして鋳物の内部に鋳巣と呼ばれる空洞の欠陥が存在する可能性がある．欠陥のサイズ，発生箇所および発生量により鋳造品の強度，耐圧性などに悪影響を与える可能性があり，高速で走行する自動車においては安全に関わる重要な問題である．しかし，製品内の鋳巣を皆無にすることは非常に困難なので，完成品は内部検査が必要となっている．

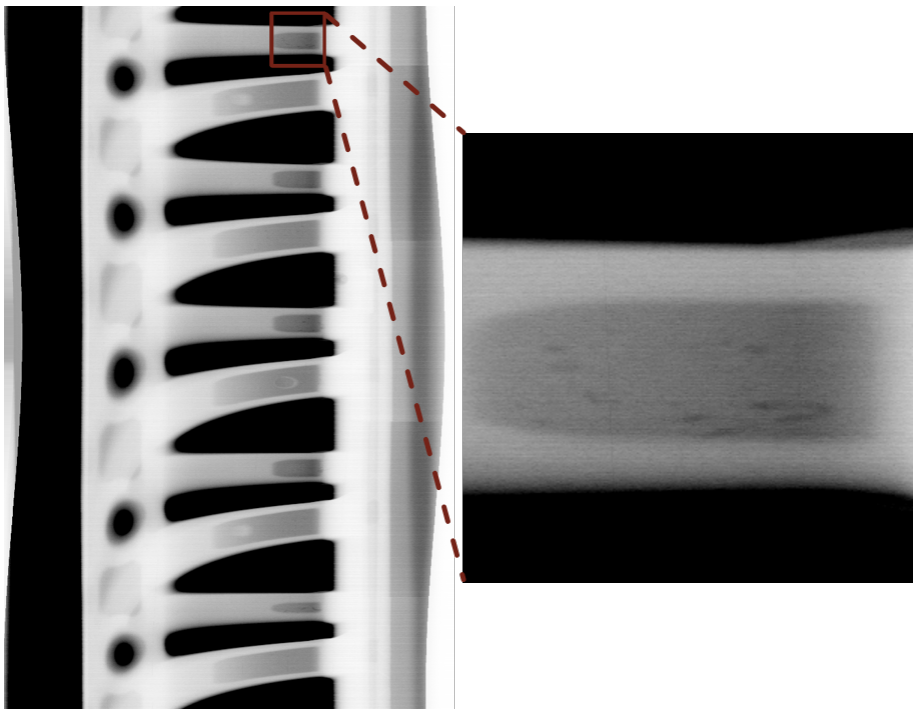


図 1.1　ホイールの X 線画像と鋳巣箇所の拡大の例．

自動車ホイールの内部検査は，従来より X 線画像（図 1.1 の左）を用いた目視検査が広く用いられている．X 線で撮影したホイール画像は白黒画像である．画像内の白い部分は金属であり，黒くなっている部分では金属の厚さが薄くなっている．そのため，図 1.1 の右のように周辺より濃度値が急に低くなるところ，つまり金属の厚さが急に薄くなるところは鋳巣である可能性が高く，作業員はそれを確認して鋳巣の位置と形状を検出する．しかし，生産ライン上での目視検査は，熟練した作業員が必要なこと，大量の部品に対して検査が必要なことなどの問題がある．さらに作業員の技量により評価結果が異なることから，品質を一定に保つことが困難であり，機械による検査の自動化のニーズが高まっている．

鋳巣検出に関する研究について，主に製品の X 線 CT 断層画像を利用した研究 [1] と超音波を利用した研究 [2] が行われた．しかし，X 線 CT 断層画像の研究では，2 値化して鋳巣を探索するため，精度にまだ問題が残っている [3]．超音波の研究では，超音波センサ真下の断面以外は検出不可であり [4]，製品全体は対応できないという欠点がある．そして，以上の 2 つの研究対象は自動車ホイールでないこともある．また，自動車ホイールの X 線画像を用いた研究 [5] も行われたが，欠陥のない X 線画像と比較するため，精度にも問題が残っている．

自動車ホイールの鋳巣検出問題の問題点として，鋳巣の定義が明確でないことが考えられる．例として，ホイールの X 線画像での鋳巣は「周辺より色が急に黒くなるところ」しか言えない．画素値がどの値より小さければ鋳巣であるかということははっきり定義できないため，検出の精度に影響を与える．近年では，ディープラーニング（深層学習）を用いた画像認識技術が進められている．これは既存の画像データに対してコンピュータを学習させる技術である．そのため，鋳巣が明確に定義されていなくても最適なモデルが作成できる可能性があり，ディープラーニングにより鋳巣検出が期待される．

しかし，自動車ホイールは様々な種類と形状があり，そして毎年に新型のホイールが開発されており，全てのホイール種に対応できるモデルの作成は非常に困難だと考えられる．そして，学習のために大量のデータが必要であるため，データ不足の問題もある．

1.2　研究目的

本論文では，人間が確認した鋳巣のデータを用い，自動車ホイールの X 線画像に Mask R-CNN[6] というディープラーニングモデルを適用し，高精度で鋳巣を自動に検出することを目指す．また，全てのホイール種への対応について，各ホイール種専用モデルの作成を解決策とし，データ不足の対策として転移学習を用いることを提案し，少ないデータ数でもデータ数が多い場合と同等もしくはそれ以上の精度を目指す．

1.3　本論文の構成

本論文の構成は次の通りである．第 2 章では，本論文で用いた基礎的知識について述べる．第 3 章では，実験で用いたホイールの X 線画像データについて説明し，データ作成およびラベリングの方法について述べる．第 4 章では，本研究で行った実験について述べる．具体的に，モデル構築，用いた評価方法を説明し，同一のホイール種の適用の実験を行い，鋳巣の検出可能性を確認する．また，複数のホイール種への対応の実験を行い，転移学習を用いることにより少ないデータ数でもデータ数が多い場合と同等もしくはそれ以上の結果が取得できることを検証する．第 5 章では，これまでに述べた内容，結果をまとめ，結論を述べる．

第 2 章

本研究の基礎となる知識

本章では，本論文で使用する基礎となる知識について述べる．2.1 節では，画像認識について紹介する．2.2 節では，畳み込みニューラルネットワークについて述べる．2.3 節では，本論文で使用した Mask R-CNN と呼ばれるモデルについて述べる．2.4 節では，転移学習について述べる．2.5 節では，画像認識分野で良く使われている COCO データセットを紹介する．

2.1 画像認識

画像認識とは，画像の中から特定の部分を抽出し，対象物を認識する技術である．人間にとっては写真に写っているのは何であるかを特定するのは容易であるが，コンピューターにとって，画像に何が写っているかを理解するのは非常に難しい．画像の内容をコンピュータに理解させるのは 1960 年代から研究が進められている．

画像認識の最初のタスクは画像分類であり，入力画像はどのカテゴリに属するかを識別する問題である．AlexNet[7] と呼ばれる畳み込みニューラルネットワーク（2.2 節を参照）のモデルが画像認識コンテストである．ImageNet Large Scale Visual Recognition Challenge（ILSVRC）で優勝した 2012 年から，画像分類に向けたディープラーニングの手法はほとんど畳み込みニューラルネットワークをベースとしているモデルであり [8, 9, 10]，画像の分類精度が人間より良いモデルもある [10, 11]．しかし，画像分類では，画像全体に対して存在するオブジェクトが何かを推論するため，結果には物体の存在する位置情報がない．さらに，物体が複数存在する場合でも上手くできないこともある．

それを解決するために，物体検出の研究が進められている．物体検出とは，入力画像に何が存在するかだけではなく，その位置も出力する技術である．また，複数の物体が存在しても対応するため，複雑な技術である．畳み込みニューラルネットワー

クをベースにして物体検出タスクに向けた様々なモデルが開発された．代表的には R-CNN[12]，Fast R-CNN[13]，Faster R-CNN[14]，You Only Look Once（YOLO）[15]，Single Shot MultiBox Detector（SSD）[16] というモデルがある．Fast R-CNN は R-CNN の改善であり，Faster R-CNN は Fast R-CNN の改善である．Faster R-CNN は YOLO と SSD より判断時間がかかるが，精度が最も良いモデルである．

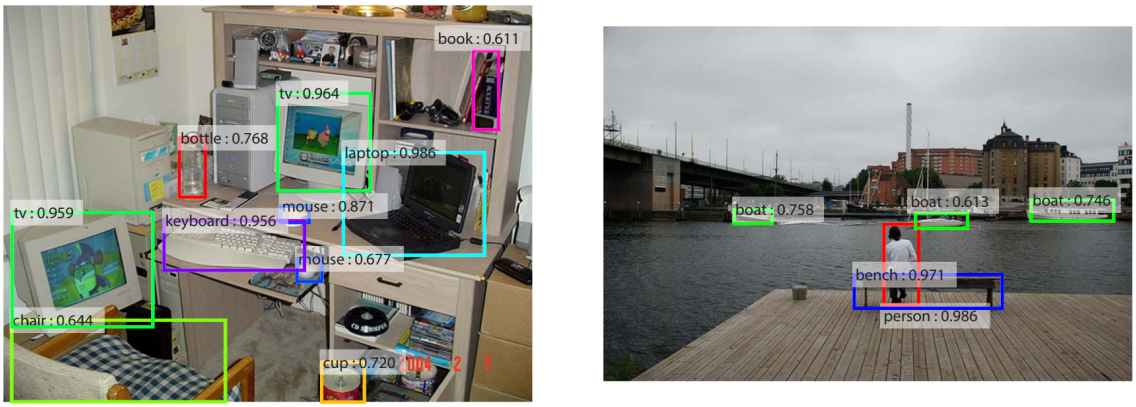
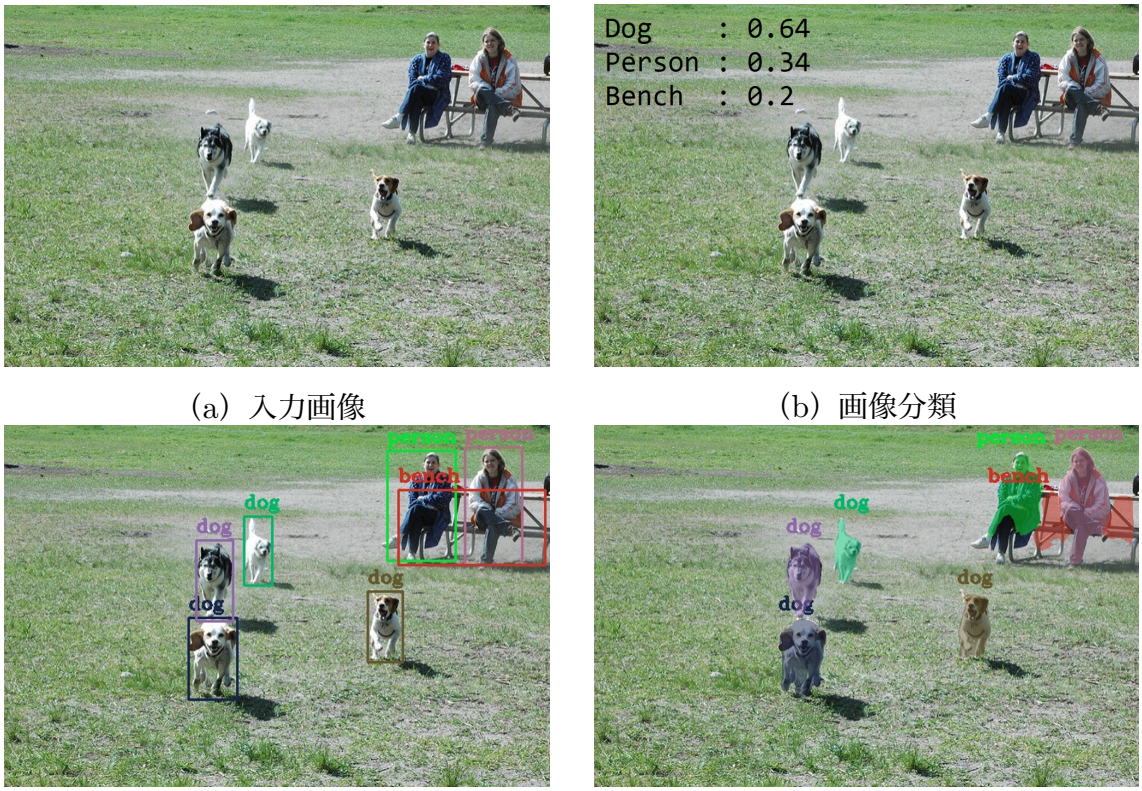


図 2.1　Faster R-CNN の結果例．各物体を囲むボックスはその物体の位置を表す．ボックスの上の文字列はその物体のクラスと存在確率である．（Faster R-CNN の論文 [14] から引用）

図 2.1 は Faster R-CNN の結果例を示す．それを見ると，Faster R-CNN は画像に対して存在する物体の位置が検出できることが確認できる．しかし，自動車ホイールの鋳巣検出問題に対して，鋳巣の位置だけではなく，鋳巣の形状も特定したいため，Faster R-CNN のような物体検出モデルではまだ不十分だと考えられる．

画像認識では，物体検出のさらに難しいタスクはインスタンスセグメンテーションである．物体検出のように，入力画像に存在する物体の位置を検出し，さらに物体が存在する領域も生成できる技術である．インスタンスセグメンテーションでは 2017 に開発された Mask R-CNN[6] と呼ばれるモデルが代表的に知られている．Mask R-CNN 以降のインスタンスセグメンテーションのモデルはほとんどそれを参考にしたものであり，それ以外には同著者が 2019 年に開発したモデル [17] もあるが，Mask R-CNN の方はまだ精度が優れている．そのため，Mask R-CNN を用いることより，自動車ホイールの鋳巣検出問題を解決することを期待する．



（a）入力画像　（b）画像分類

（c）物体検出　（d）インスタンスセグメンテーション

図 2.2　画像分類，物体認識，インスタンスセグメンテーションの比較．

2.2　畳み込みニューラルネットワーク

畳み込みニューラルネットワーク（Convolutional Neural Network: CNN）は，画像や動画認識に広く使われているモデルである．畳み込みニューラルネットワークは主に畳み込み層，プーリング層，全結合層，合計 3 つの部分から構成される．畳み込み層の入力は画像もしくは特徴マップであり，その入力に畳み込み処理を行って特徴を抽出し，設定した活性化関数で処理して出力する．プーリング層は通常畳み込み層の直後に設定され，畳み込み層とセットになり，畳み込み層の出力をダウンサンプリングすることにより，モデルのパラメータ数を減少する処理と共に，重要な特徴を抽出する．全結合層は畳み込み層とプーリング層で検出された特徴マップを分類する．畳み込みニューラルネットワークは通常画像分類問題で適用される．

2.2.1　畳み込み層

畳み込み層は CNN で最も重要な部分であり，基本構造は図 2.3 に示す．サイズ $h \times w \times d$（d はチャネル数）の画像を入力として考える．畳み込み層はサイズ $h_f \times w_f \times d$ のフィルタで演算する．ただし，畳み込み処理のため，フィルタと入力画像のチャネル数は一致しないといけない．

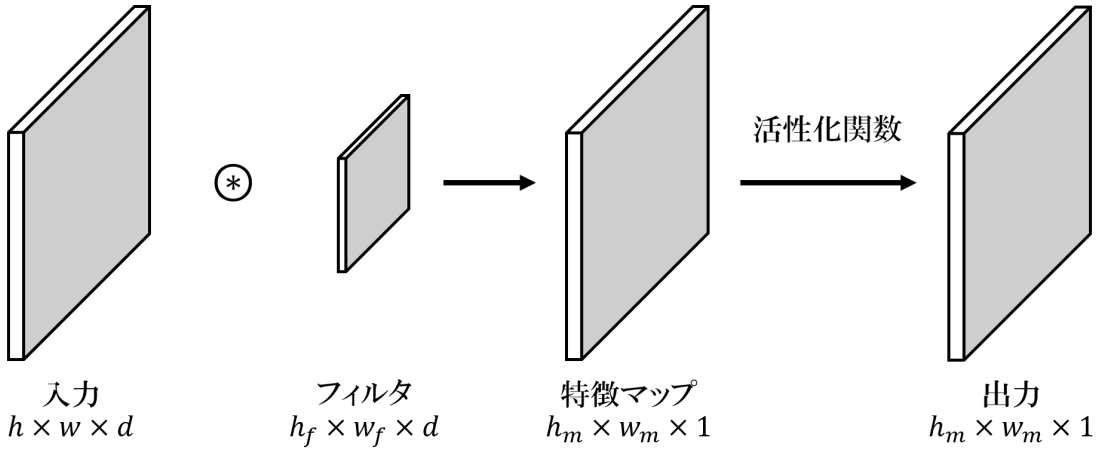


図 2.3　畳み込み層の例．（畳み込み演算は ⊛ で表す）

畳み込み演算は図 2.4 のように，フィルタに対応する入力の要素を乗算し，その和を計算する．この計算でフィルタ窓を入力の上に移動しながら全ての入力の要素に対して行い，$h_m \times w_m \times 1$ の特徴マップが得られる．

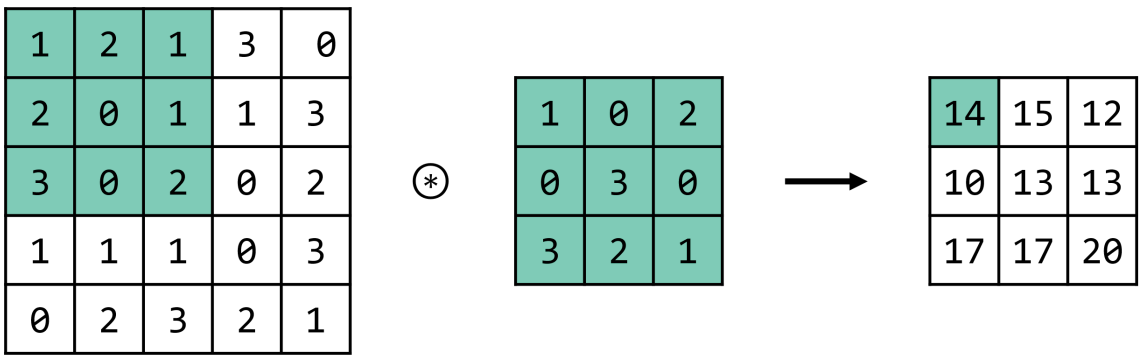


図 2.4　畳み込み演算の例．（畳み込み演算は ⊛ で表す）

特徴マップのチャネル数はフィルタ数により変わる．例として，畳み込み層のフィルタ数が 1 の場合，特徴マップのサイズは $h_m \times w_m \times 1$ であり，フィルタ数が 6 の場合，特徴マップのサイズは $h_m \times w_m \times 6$ となる．

また，特徴マップのサイズは入力のサイズ，フィルタのサイズ，フィルタの窓の移動幅である stride（ストライド），そしてゼロパディングをするかどうかで決められる．通常，特徴マップと入力のサイズが同じになるように，ゼロパディングを行う（図 2.5）．そのため，畳み込み層はフィルタのサイズ，フィルタ数，ストライドだけで表すことが一般的である（例 7×7, 64, stride 2）．

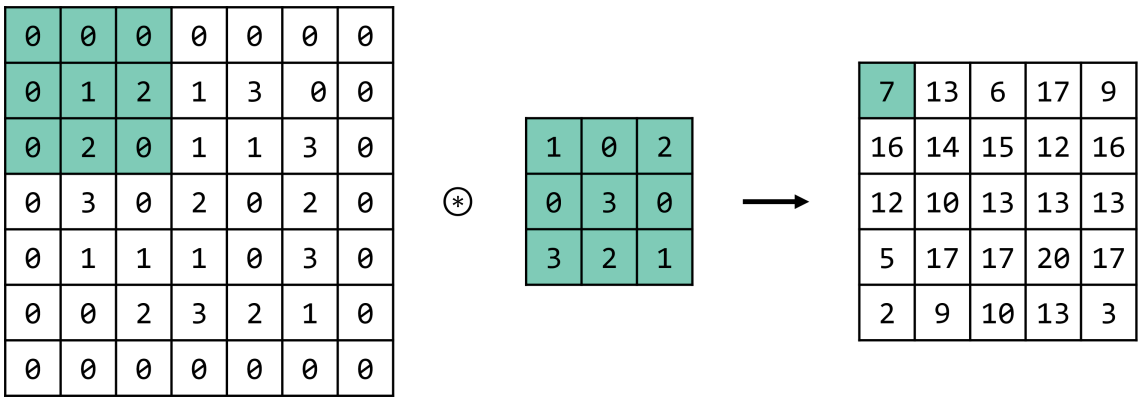


図 2.5　ゼロパディングの例．5×5 の入力画像にゼロパディングを行い，3×3 のフィルタで畳み込み処理して 5×5 の出力が得られる．（畳み込み演算は ⊛ で表す）

最後に，得られた特徴マップは非線形性を付加するために活性化関数で処理を行う．現在は ReLU（式（2.1）と図 2.6）の関数が良いと言われており [18]，それを利用するのが一般的である．

$$\mathrm{Relu}(x) = \max(0, x) \tag{2.1}$$

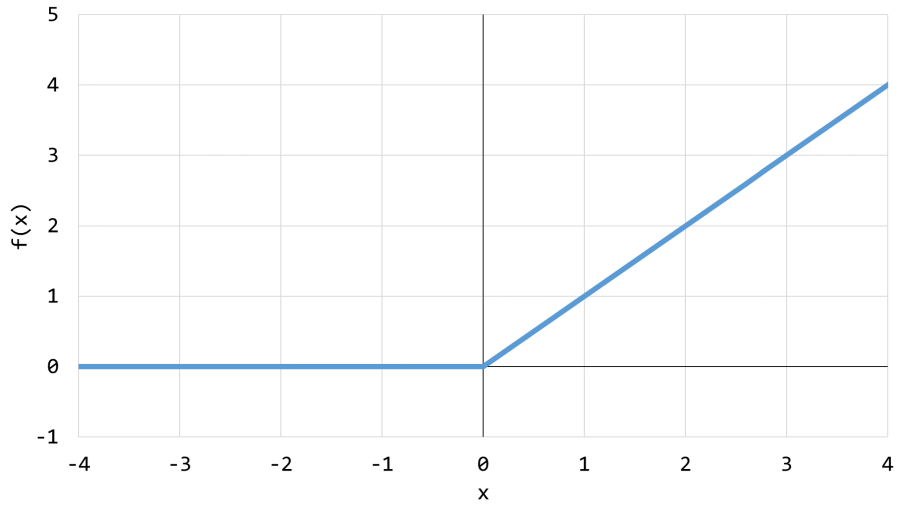


図 2.6　ReLu 関数.

2.2.2　プーリング層

プーリング層の目的は，計算の負荷，メモリ使用量，モデルのパラメータ数の削減であり，畳み込み層の出力をダウンサンプリング処理する．プーリング層は畳み込み層と同様にフィルタとストライドを設定し，入力に対してフィルタを適用して出力する．プーリング処理では各範囲の最大値を出力する max pooling（最大プーリング）が良く使われる．また，プーリング層の出力のサイズが入力の半分になるように stride は通常 2 で設定される．

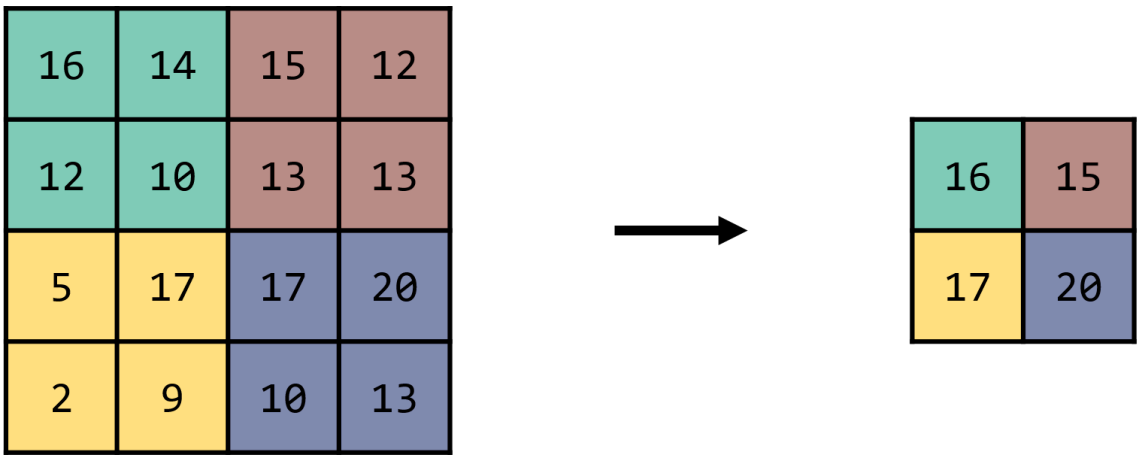


図 2.7　2 × 2 max pooling, stride 2 の例.

図 2.7 は 2 × 2 max pooling, stride 2 の例である．図のように，プーリング処理は入出力のチャネル数は変化しない．また，プーリング層のフィルタはパラメータを持たず，学習により変化する重みもない．

畳み込みニューラルネットワークでは，複数の畳み込み層とプーリング層で構成される部分が特徴検出器の役割を担当する．特徴検出器の出力は入力画像の最終特徴マップであり，分類器の入力となる．

2.2.3　全結合層

全結合層は，図 2.8 のように，従来の多層パーセプトロンのニューラルネットワークと同じ構造である．ニューロンが次の層の全てのニューロンに接続するため，全結合層と呼ばれる．

多層パーセプトロンであるため，全結合層の入力は 1 次元の行列となり，特徴検出器の出力を 1 次元に変換して入力とする．入力に重みをかけて計算して出力する．全結合層は出力が正解ラベルに対する 1 次元行列であり，畳み込みニューラルネットワークの分類器の役割を担当する．

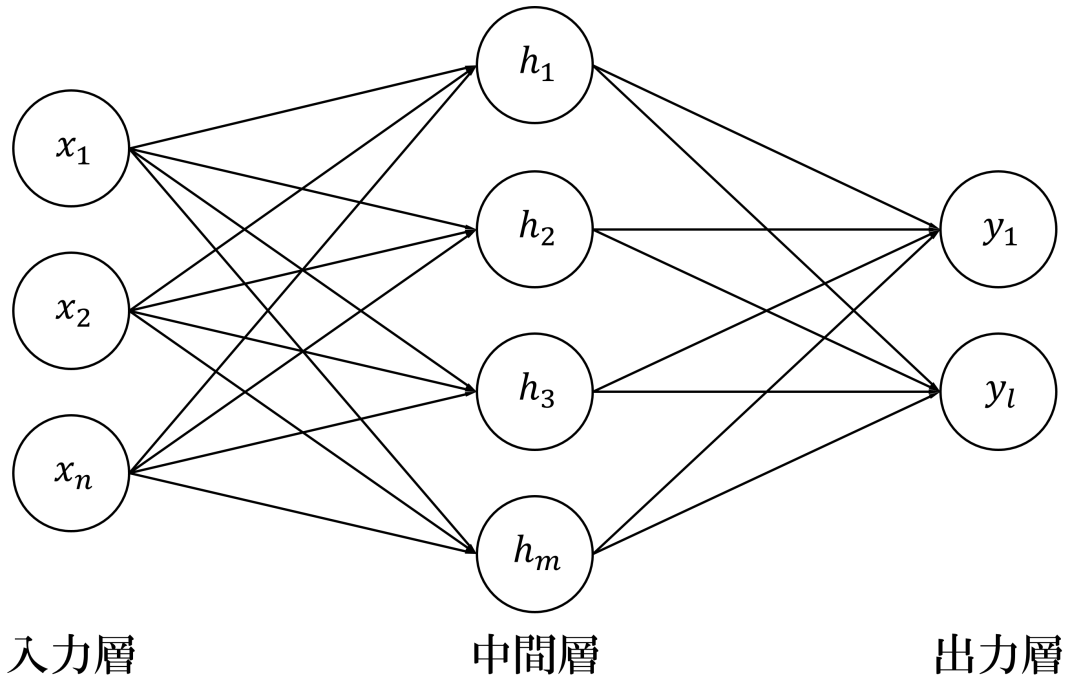


図 2.8　全結合層の構造例.

2.3　Mask R-CNN

Mask R-CNN[6] は 2.1 節で言及した物体検出タスクに向けた Faster R-CNN に物体のマスク（物体が存在する領域）を生成する部分を追加して改善したモデルである．したがって，Faster R-CNN と同様に入力画像に存在する物体が検出できる可能性がある．さらに，改善した部分により物体のマスクも生成できる（図 2.9）．

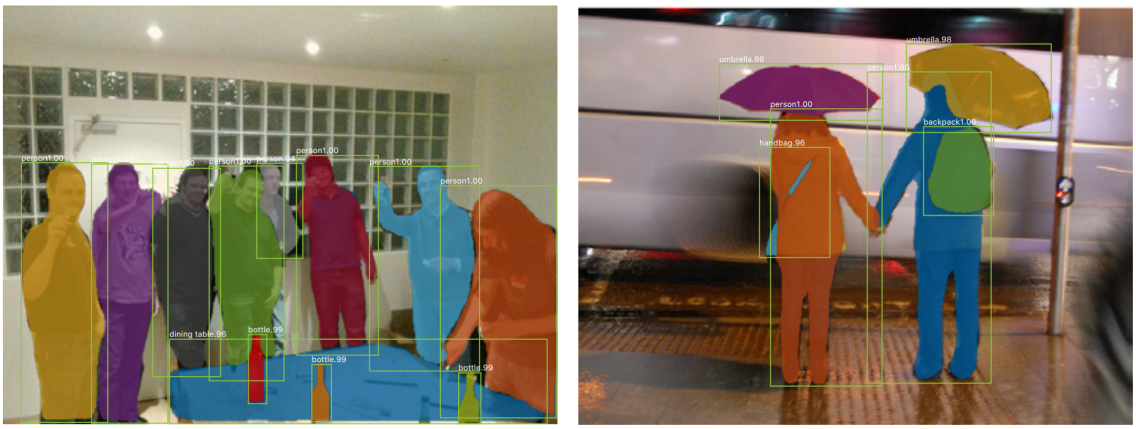


図 2.9　Mask R-CNN の結果例．（Mask R-CNN の論文 [6] から引用）

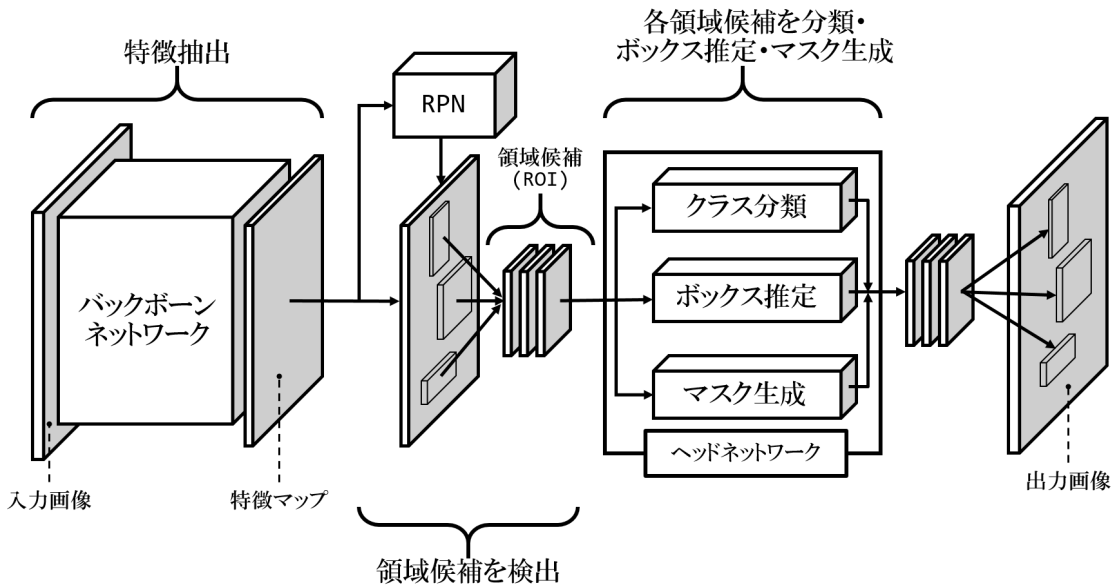


図 2.10　Mask R-CNN の構造．RPN は Region Proposal Network の省略．

Mask R-CNN の構造は図 2.10 に示すとおりであり，「バックボーンネットワーク」，「Region Proposal Network（RPN）」，そして「ヘッドネットワーク」の 3 つのブロックに分けられる．バックボーンネットワークは入力画像から特徴マップを抽出する．RPN はその特徴から物体が存在する位置（バウンディングボックス）を提案する．ヘッドネットワークは RPN が提案した全てのバウンディングボックスに対してクラス分類を行い，位置を推定し，マスクを生成して出力する．バックボーンネットワーク，RPN，ヘッドネットワークの詳細の構造は以下の 2.3.1 節，2.3.2 節，2.3.3 節で解説する．

2.3.1　バックボーンネットワーク

バックボーンネットワークは主に複数の畳み込み層から構成され，入力画像の特徴を抽出する役割を担当する．Mask R-CNN の論文では 50 層または 101 層の畳み込み層から構成されるネットワークを使用していた．このネットワーク自体は画像分類タスクでよく知られている ResNet[10] と呼ばれる畳み込みニューラルネットワークの特徴検出器であり，画像分類の精度は人間より良い結果が得られたモデルである (2.1 節)．

また，異なるスケールの物体でも良い検出精度が得られるために，バックボーンネットワークでは Feature Pyramid Network（FPN）[19] も導入されている．FPN は，横方向の接続を備えたトップダウンアーキテクチャ（図 2.11）を使用し，単一スケールの入力から複数スケールの特徴マップを取得できる．

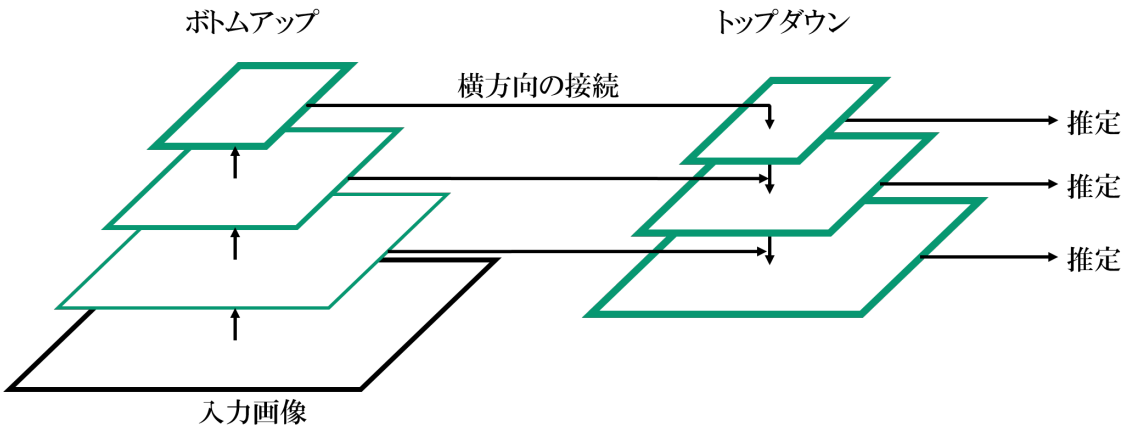


図 2.11　Feature Pyramid Network.

2.3.2　Region Proposal Network

Region Proposal Network（RPN）[14] の入力は任意のサイズの画像であり，その画像に物体が存在する領域（バウンディングボックス）と物体である確率（スコア）をセットで出力する．RPN の出力は候補領域（Region of Interest: RoI）と呼ぶ．入力画像に物体が複数存在する場合は複数のバウンディングボックス・確率のセットが出力される．Mask R-CNN では，RPN の入力はバックボーンネットワークで得られた特徴マップである．

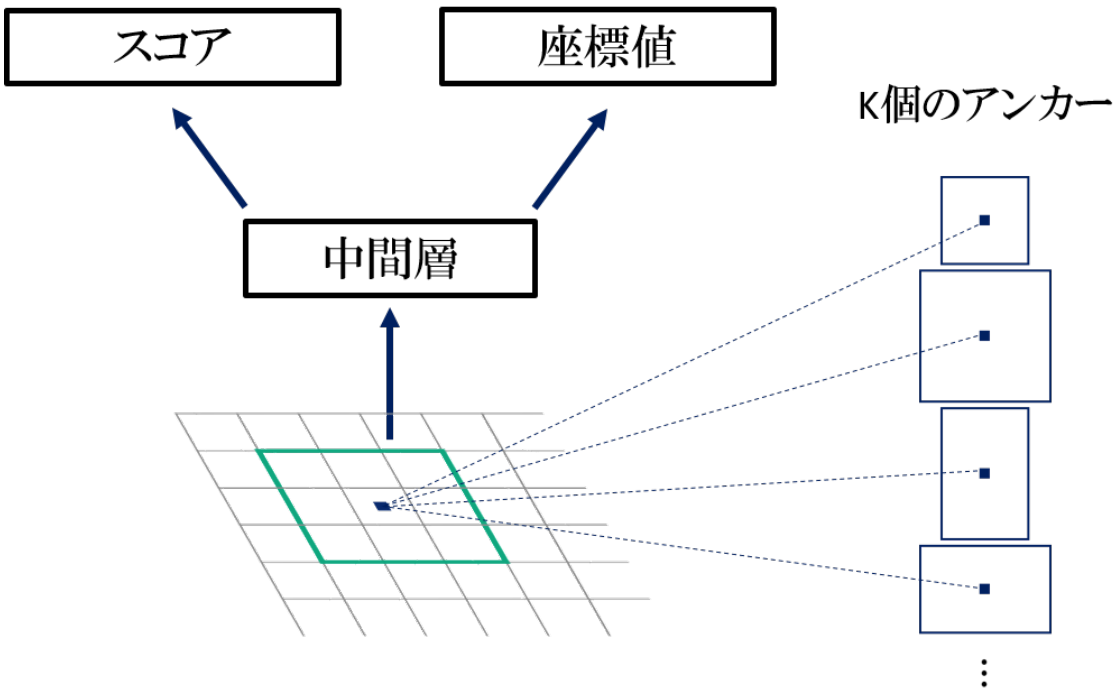


図 2.12　Region Proposal Network.

候補領域を生成するために，入力にスライド窓を移動する．この窓は入力の $n \times n$ の空間窓を中間層でより低次元の特徴マップに減少させ，この特徴マップは 2 つの全結合層（バウンディングボックス回帰層（reg）と分類層（cls））に入力される．この構造は $n \times n$ の畳み込み層と 2 つの 1×1 の畳み込み層から構成される．

入力画像では，同じ領域に複数の物体が存在する可能性があるため，1 つのスライド窓の位置では，複数の候補領域を提案する．1 つの窓での最大候補数を k と仮定し，その窓でサイズが異なる k 個のボックス（アンカーと呼ぶ）を RPN の入力として同時に処理する．その時に，バウンディングボックス回帰層の出力は k 個のアンカーの座標を表す $k \times 4$ の行列であり，分類層の出力は入力アンカーに対して物体であるかどうかのスコアを表す $k \times 2$ の行列である．アンカーの中心はスライド窓の中心であ

り，サイズはスケール（ピクセル数）とアスペクト比（幅/高さ）で決められる．

RPN を学習するには，各アンカーにバイナリラベル（物体か物体でないか）を作成する．ポジティブアンカーは 1 つのラベルボックスに対して IoU（4.2.1 節を参照）が最も高いもしくは任意のラベルボックスに対して IoU が 0.7 以上のアンカーである．ネガティブアンカーは全てのラベルボックスに対して IoU が 0.3 以下のアンカーとする．それ以外のアンカーは学習に使用しない．

上記の定義で，RPN での損失関数は式（2.2）で表す．

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{2.2}$$

ここで，i はバッチでのアンカーのインデックスであり，p_i はそのアンカーが物体であることを推定した確率である．p_i^* はラベルの確率であり，ポジティブアンカーなら 1，ネガティブなら 0 である．t_i は推定したバウンディングボックスの位置を表すパラメータ化したベクトルであり，t_i^* はポジティブアンカーに対するラベルボックスのものである．分類の損失 $L_{cls}(p_i, p_i^*)$ はクロスエントロピーを使用し，バウンディングボックス回帰の損失は $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ とする．ただし，R は [13] で定義した Smooth L1 と呼ばれる損失関数（式（2.3）と図 2.13）である．また，$p_i^* L_{reg}$ の意味はバウンディングボックス回帰の損失はポジティブアンカーのみ（$p_i^* = 1$）で計算され，ネガティブアンカーの場合は 0 となる（$p_i^* = 0$）．2 つの損失は N_{cls} と N_{reg} で正規化され，λ でバランスされる．N_{cls} はバッチサイズであり，N_{reg} はアンカーの数である．

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if} |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{2.3}$$

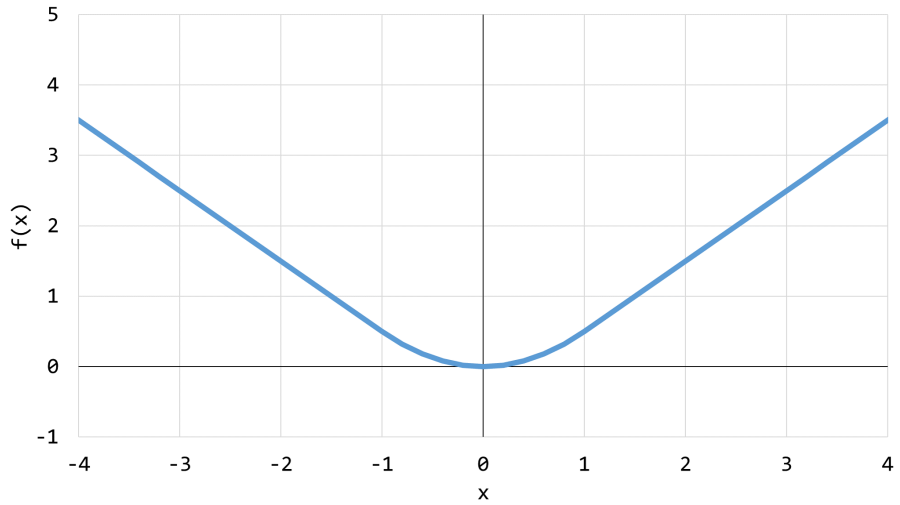


図 2.13　Smooth L1 の損失．

バウンディングボックス回帰では，以下のようにパラメータ化する．

$$\begin{array}{ll} t_x = (x - x_a)/w_a & t_y = (y - y_a)/h_a \\ t_w = \log(w/w_a) & t_h = \log(h/h_a) \\ t_x^* = (x^* - x_a)/w_a & t_y^* = (y - y_a)/h_a \\ t_w^* = \log(w^*/w_a) & t_h^* = \log(h^*/h_a) \end{array} \tag{2.4}$$

ただし，x，y，w と h はボックス中心の座標とその幅，高さである．x，x_a，x^* はそれぞれ推定したボックス，アンカーボックスとラベルボックスである（y，w，h も同様）．これはアンカーボックスを近くのラベルボックスへの回帰だと考えられる．

2.3.3　ヘッドネットワーク

Mask R-CNN のヘッドネットワークは複数の畳み込み層と全結合層から構成され，RPN で提案された全ての候補領域に対してクラス分類，位置推定，マスク生成を行う．候補領域の特徴を $m \times m$ の特徴マップに変換し，ヘッドネットワークはその特徴マップを 2 つの出力に分ける全結合層に入力することにより物体のボックスおよび物体クラスとその確率が得られる．また，各候補領域に対してクラス分類，ボックス推定と並行して複数の畳み込み層からなるネットワークでマスク生成を行う．クラス分類と位置推定の損失は RPN の損失と同様であり，マスク生成の損失はバイナリクロスエントロピーを使用した．

2.4　転移学習

近年，インターネットの普及により学習データの増加と共に，コンピュータなどの処理速度が高速化され，画像認識分野での結果は益々良くなっている．良く使われているデータセットは 120 万枚，1000 クラスの ImageNet データセットであり，そのデータセットを使用する画像認識コンテストの ILSVRC で優勝したモデルがたくさんある．それらのモデルは，ほとんど複数の層からなるディープラーニングによるモデルであり，入力側の層は畳み込み層とプーリング層で構成され，最後にある層が全結合層のモデルは一般的である．最後の層の出力サイズはクラス数と同じため，最後から 2 番目の層の出力は入力の特徴マップであり，その最後の層は分類器だと考えられる（複数の全結合層の場合はその前の層の出力が特徴マップである）．ディープラーニングのモデルは特徴検出器と分類器を同時に学習するため，良い精度が取得できる．複数の層から構成されるモデルなので，ImageNet データセットの 120 万枚を全部学習するには時間が非常にかかる．他のデータセットを使用する場合は，数が多いデータセットはあまりないため，最初からモデルを構築して学習することはしない．その代わりに，学習済みのモデルから転移学習という技術を使用する．

転移学習とは，ある領域で学習したことを別の領域に役立たせ，適合させる技術である．深層学習では，既存の学習済モデルの一部もしくはすべての重みを自分のモデルに使用することである．転移学習を導入すれば，学習時間を短縮することができ，データが少なくても高い精度のモデルが得られる利点がある．例として，自動車が認識できるモデルはトラックを認識しようとするときに応用できる可能性がある．自動車が認識できるモデルは特徴検出器と分類器が自動車に対応している．トラックは自動車と同様に乗り物であるため，自動車に近い特徴を持っていると考えられ，自動車のモデルの重みを少量のトラックのデータで微調整すればトラックが認識できるようになる．

2.5　COCO データセット

COCO（Common Objects in Context）データセット [20] は物体検出，セグメンテーション，キーポイント，キャプションに向けたラージスケールのデータセットであり，マイクロソフトが 2015 年に提出されたデータセットである．全部 80 のカテゴリラベル，物体のインスタンス 1500 万個とデータ数は 20 万枚以上がある．物体認識，セグメンテーションタスクではデータ数が最も多く良く使われているデータセットである．Faster R-CNN，Mask R-CNN などのモデルは COCO データセットを使用した．そのため，Mask R-CNN を使用するときは，自分のデータを COCO データセットのフォーマットに変換することが一般的である．

（a）画像　　（b）セグメンテーションのラベル

図 2.14　COCO データセットの例.

第 3 章

データ作成

本章では，本論文で使用したデータセットについて述べる．3.1 節では，ホイールのデータとデータ数について述べる．3.2 節では，学習のためのデータ作成について説明する．

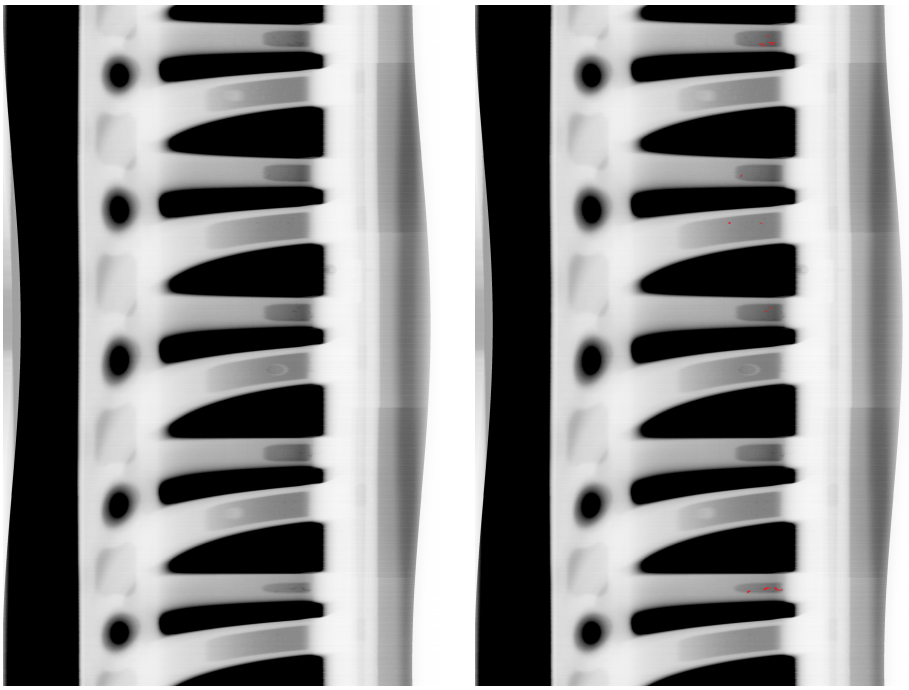
3.1 ホイールデータ

3.1.1 ホイール画像

本論文で使用したホイール画像は 1.1 節に言及したようにホイールの X 線画像であり，図 3.1 に示す．図 3.1（a）はホイールから撮影して加工した画像（フル画像）であり，図 3.1（b）はその画像に対して鋳巣箇所に人間が赤マークを入れたもの（マークフル画像）である．このようなフル画像とマークフル画像は 1 つのセットとなり，フル画像は学習用とテスト用とし，マークフル画像はラベルにする．

3.1.2 ホイール種とデータ数

ホイールは様々な種類がある．本論文のホイールデータでは，全部 21 種類のホイール種がある．ホイールデータ内では，データ数が多いのが BKK，BMC，AZN，BGD，AZQ，BMB と呼ばれるホイール種であるため，実験ではそれらのホイール種で行った．また，学習とテストのため，ホイールデータを学習用とテスト用に分け，割合は学習用，テスト用それぞれ 8 割，2 割である．それぞれのデータ数は表 3.1 に表し，形状は図 3.2 に示す．

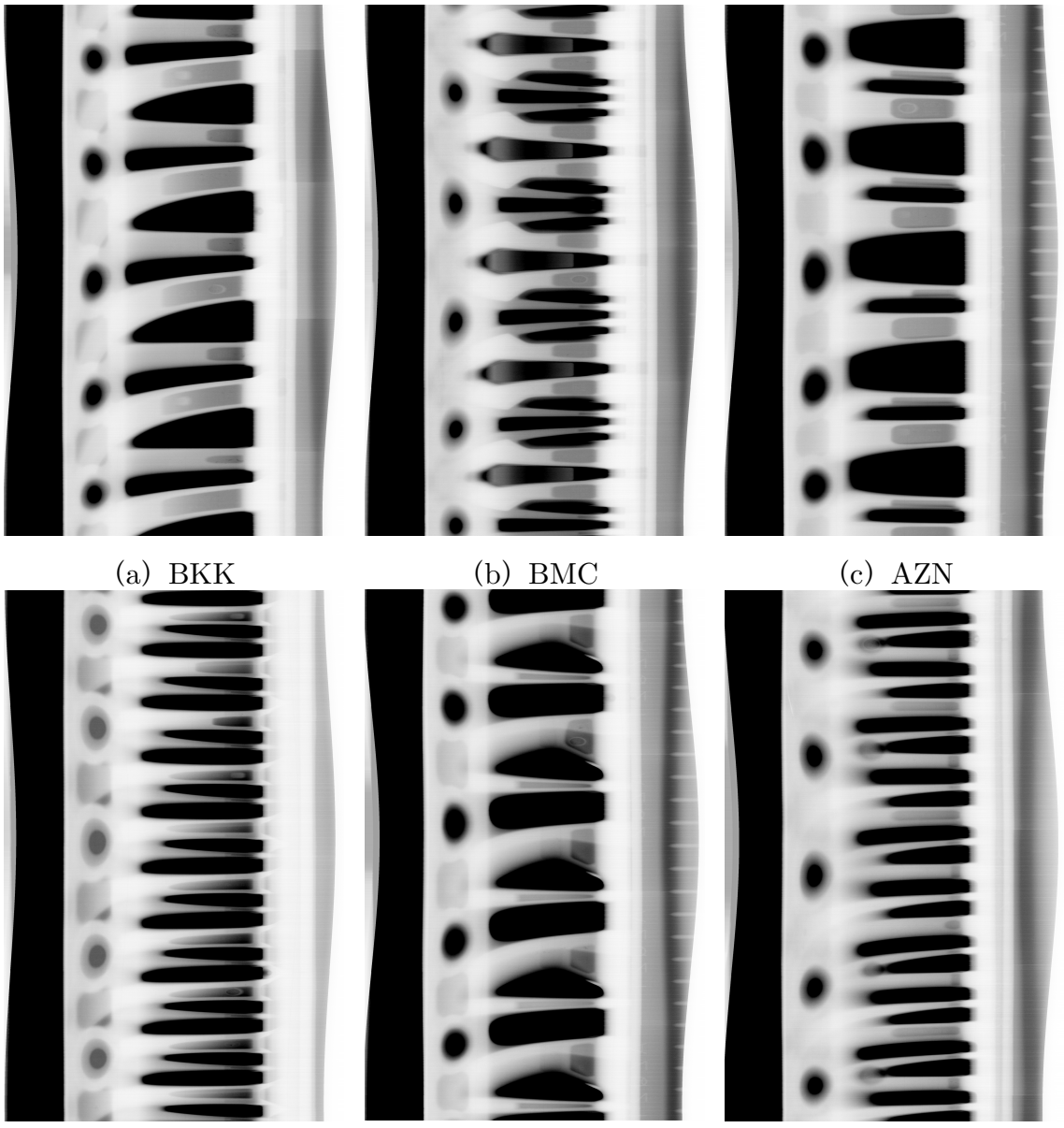


(a) ホイールのフル画像　　(b) マークされたフル画像

図 3.1　ホイールのフル画像.

表 3.1　ホイール種とデータ数.

ホイール名	データセット数	学習用	テスト用
BKK	300	240	60
BMC	260	208	52
AZN	180	144	36
BGD	155	124	31
AZQ	150	120	30
BMB	150	120	30



(a) BKK　(b) BMC　(c) AZN

(d) BGD　(e) AZQ　(f) BMB

図 3.2　本論文で使用するホイール種の形状.

3.2　データ作成

図 3.1 (a) のようなフル画像のサイズは 4000×2560 である．このサイズでは，どのモデルでも学習が困難であるため，フル画像より小さい画像に切り出し，COCO データセットのフォーマットに変換してラベリングすることが必要である．

3.2.1　フル画像の切り出し

フル画像の切り出しについては，フル画像に $h_c \times w_c$ のスライド窓を移動させ，その窓に対するフル画像の部分を出力してデータにする．しかし，1 枚のフル画像に存在する鋳巣の数は非常に少ないため，そのスライド窓の縦と横の移動幅はそれぞれ $h_c/2$ と $w_c/2$ サイズの半分とし，図 3.3 のように切り出す．フル画像にテストを行う際も，データ作成のように移動幅 $h_c/2$ と $w_c/2$ を考慮してスライド窓を移動しながら，その窓に対してテストを行う．これは端部分に存在する鋳巣の検出漏れを防止するためである．スライド窓のサイズは 128×128 にした．

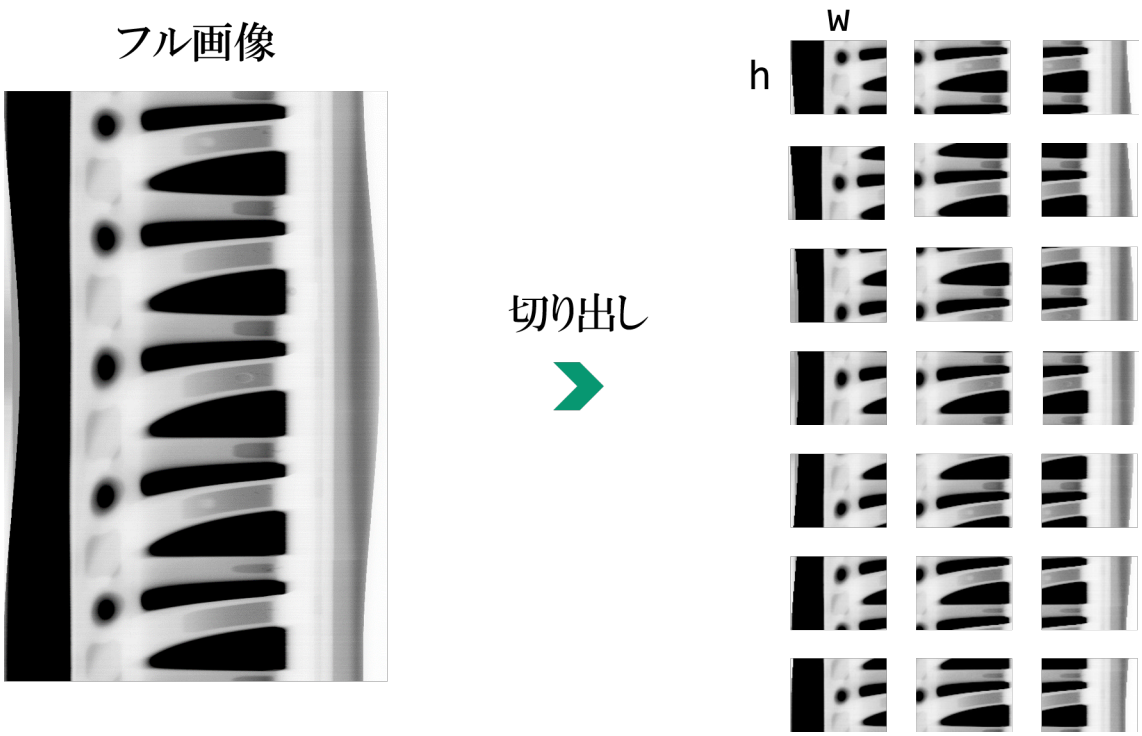


図 3.3　フル画像の切り出し．

上記の方法により，1 枚のフル画像から 2418 枚のデータ画像が切り出せる．マークフル画像も同様に $h_c \times w_c$ の画像（マーク画像）に切り出し，ラベルとして使用する．ホイール種それぞれのデータ画像の数は表 3.2 に表す．

表 3.2　学習で使用するホイール種のデータ画像数.

ホイール名	データセット数	鋳巣なしの データ画像の数	鋳巣ありの データ画像の数
BKK	240	562212	18108
BMC	208	497936	5008
AZN	144	343124	5068
BGD	124	296227	3605
AZQ	120	286862	3298
BMB	120	288272	1888

3.2.2　画像データのラベリング

COCO データセットのフォーマットに変換するために pycococreator[21] というツールを使用する．そのツールの仕様により，ラベルのために検出したい物体のマスクを作成する必要がある．そのため，各データ画像に対してホイール部分と鋳巣部分のバイナリマスクを作成する．バイナリマスクはその画像と同じサイズの白黒画像である．バイナリマスクには，対象の物体は白い画素，背景は黒い画素で表す．図 3.4 のように，鋳巣はデータ画像に対するマーク画像の赤マークを利用し，ホイール部分は画素値 30 以上にした．

COCO データセットのフォーマットに変換する際には，各物体は任意の形で存在するが，バイナリマスクの情報を保存するために，pycococreator ツールの使用により，そのマスクを多角形に変換する．そのため，画像の端部分に存在する非常に小さい赤マークは変換後なくなることがある．これはデータ切り出しの際に，鋳巣が切られ，データ画像に非常に小さい赤マークが残されるためである．このような画像は数が少なく，データの鋳巣なしと鋳巣ありの割合は関係があるが，Mask R-CNN の精度には悪影響しないと考えられる．

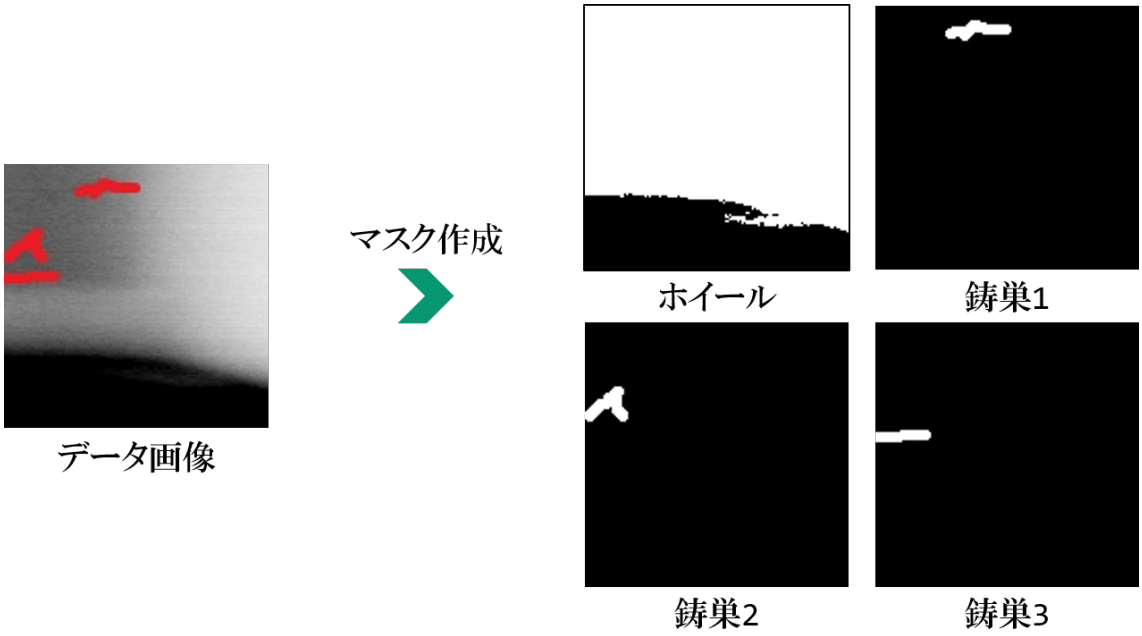


図 3.4　ホイールと鋳巣のバイナリマスクの作成．ホイール部分の画像の枠は観測のために追加した．

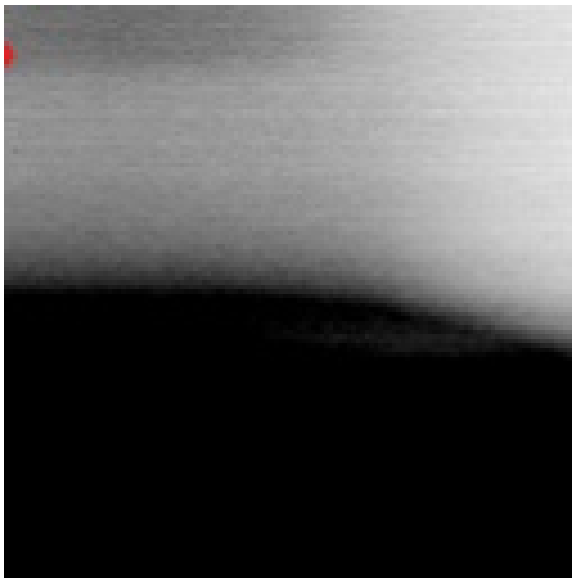


図 3.5　データ切り出しの際に，鋳巣が切られて端部分に小さい赤マークが残されるデータ画像．データフォーマット変換ツールの使用により，このようなラベルはデータフォーマット変換後なくなる．

第 4 章

実験

本章では，本論文で行った実験について述べる．4.1 節では，モデル構築について述べる．4.2 節では，用いた評価方法について説明する．4.3 節では，Mask R-CNN を用いて同一のホイール種に適用し，自動車ホイールの X 線画像から鋳巣検出の可能性を確認する実験について述べる．4.4 節では，同一のホイール種で学習したモデルを用い，複数のホイール種にテストを行い，学習に使用しないホイール種の鋳巣検出の可能性を確認する実験について述べる．4.5 節では，転移学習を用いて少ないデータ数でもデータが多いモデルと同等もしくはそれ以上の精度を取得する実験について述べる．4.6 節では，転移学習で必要となるデータ数を検討する実験について述べる．

4.1　モデル構築

本論文で使用したバックボーンネットワークの構造は 50 層の ResNet50 であり，ネットワーク構造は表 4.1 に示す．RPN のスライド窓のサイズは 3×3 であり，その中間層のサイズは 512 にした．また，(8, 16, 32, 64) の 4 つのスケールと (0.5, 1, 2) の 3 つのアスペクト比を使用し，1 つのスライド窓では 12 個のアンカーを生成するように設定し，ポジティブアンカーとネガティブアンカーの IoU の閾値はそれぞれ 0.7, 0.3，バランス重み $\lambda = 1$ にした．ヘッドネットワークは図 4.1 に示す．モデル構築は文献 [22] を利用した．

表 4.1　バックボーンネットワークの構造.

層名	構造
conv1	$7 \times 7,\ 64,\ \text{stride } 2$
conv2_x	$3 \times 3,\ \text{max pooling},\ \text{stride } 2$ $\begin{bmatrix} 1 \times 1,\ 64 \\ 3 \times 3,\ 64 \\ 1 \times 1,\ 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1 \times 1,\ 128 \\ 3 \times 3,\ 128 \\ 1 \times 1,\ 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1 \times 1,\ 256 \\ 3 \times 3,\ 256 \\ 1 \times 1,\ 1024 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 1 \times 1,\ 512 \\ 3 \times 3,\ 512 \\ 1 \times 1,\ 2048 \end{bmatrix} \times 3$

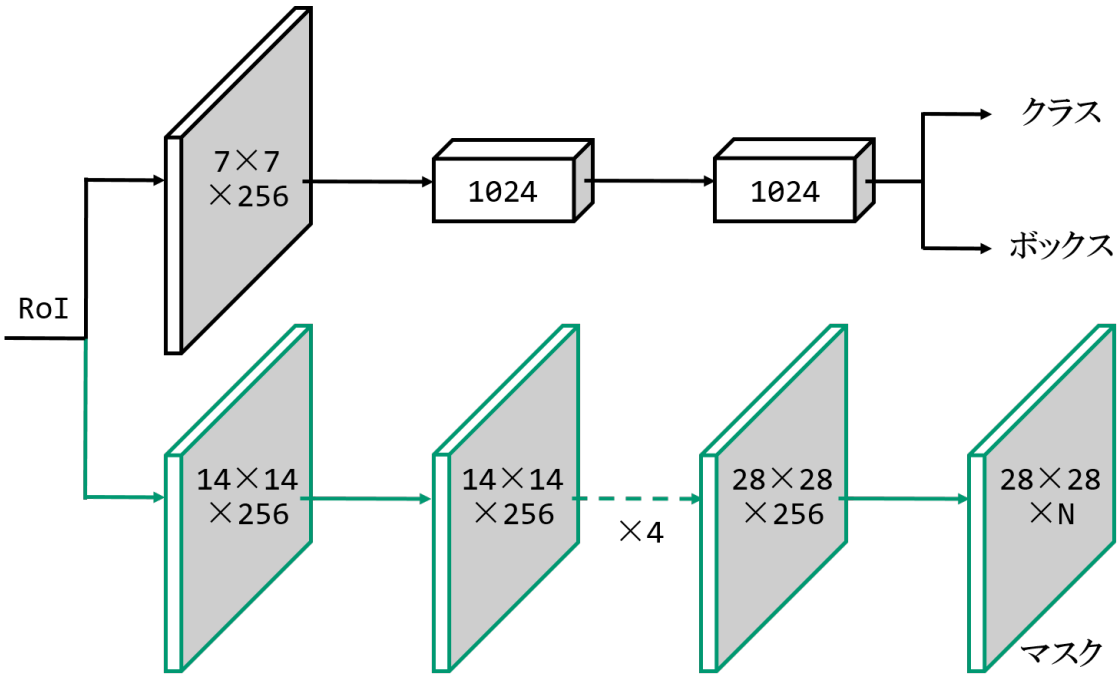


図 4.1　ヘッドネットワークの構造．矢印は畳み込み層もしくは全結合層を示す．出力の畳み込み層は 2×2 であり，残り全ての畳み込み層は 3×3 である．隠れ層での活性化関数は ReLU であり，N はクラス数である．

4.2　評価方法

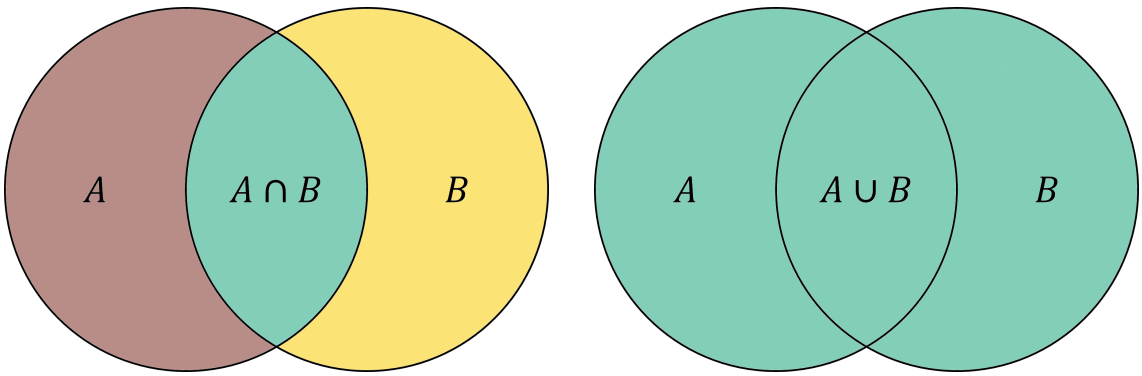
Mask R-CNN を用いることより，入力画像に鋳巣が存在するかどうかだけではなく，鋳巣の位置，形状も特定できる．そのため，領域の一致度の評価が必要となる．本論文では，下記の IoU および Precision と Recall を用いて評価した．

4.2.1　IoU

IoU（Intersection over Union）は 2 つのセットの類似度を表す指数であり，2 つセットの和集合に対して共通集合の割合で計算する．2 つのセットを A，B とすると，IoU は式（4.1）のように計算する．IoU の範囲は 0 ～ 1 であり，高い方が良い．ただし，A と B 両方が空集合の場合は IoU = 1 となる．

$$\mathrm{IoU}(A, B) = \frac{A \cap B}{A \cup B} \tag{4.1}$$

図 4.3 は物体検出に対して IoU の計算方法である．本論文では，人間が目視で入れた赤マークをラベルとし，1 枚のフル画像に対してラベルマスクとモデルが生成したマスク（生成マスク）との IoU を計算してテスト用のデータセットの平均で評価する．



（a）共通集合　　（b）和集合

図 4.2　共通集合と和集合を用いる A と B 2 つのセットの IoU の計算方法．

また，結果観測のため，ラベルマスクが検出された割合である IoA と，生成マスクが正解した割合である IoB も計算して評価する．IoA，IoB の計算方法はそれぞれ式（4.2），式（4.3）に示す．

$$\mathrm{IoA}(A, B) = \frac{A \cap B}{A} \tag{4.2}$$

$$\mathrm{IoB}(A, B) = \frac{A \cap B}{B} \tag{4.3}$$

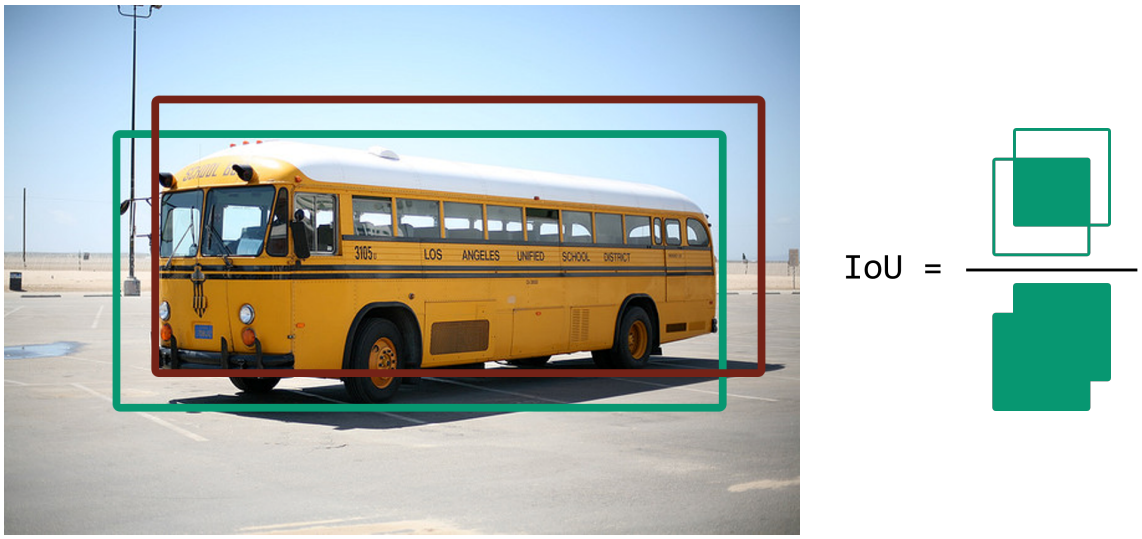


図 4.3　画像に対して IoU の計算方法．緑のバウンディングボックスはラベル，赤のバウンディングボックスは結果で表す．

4.2.2　Precision と Recall

IoU を用いることにより，ラベルマスクと生成マスクの類似度を計算することができ，その類似度が 1 に近づいたら良いと分かるが，値の差の意味がイメージしにくい．例として，テスト用データセットの平均 IoU が 0.3 と 0.4 のモデルがあるとし，0.4 のモデルは 0.3 のモデルより良い精度を持っているが，具体的にどう良くなるか，その 0.1 の差は何を意味するかは分からない．そのため，IoU と共に，Precision と Recall の評価指標も追加して評価方法として使用する．Precision はモデルが検出した鋳巣に対して正解した鋳巣（正解鋳巣）の割合を表し，Recall はラベルの鋳巣に対するモデルの正解鋳巣の割合であり，式（4.4）と式（4.5）のように計算する．Precision と Recall の範囲も 0 ～ 1 であり，高い方が良い．

$$\text{Precision} = \frac{\text{正解鋳巣の数}}{\text{モデルが検出した鋳巣の数}} \tag{4.4}$$

$$\text{Recall} = \frac{\text{正解鋳巣の数}}{\text{ラベルの鋳巣の数}} \tag{4.5}$$

ただし，モデルの正解鋳巣の定義は，閾値 $T(0 \leqslant T \leqslant 1)$ を決め，生成マスクとそれに対するラベルマスクとの IoU を計算し，IoU が閾値 T 以上なら正解鋳巣とする．本論文では，Precision と Recall の閾値 T は 0.2 にした．フル画像 1 枚に対して全部の窓のテストが終わったら，5 つの評価指標を計算し，最後に全てのテストのフル画像の平均を計算して評価する．

4.3　同一ホイール種への適用実験

本実験は，データ数がもっとも多いホイール種である BKK ホイールを用い，Mask R-CNN を適用して自動車ホイールの X 線画像からの鋳巣検出の可能性を確認する実験である．

4.3.1　条件

BKK ホイールのデータ画像の中に，鋳巣ありを 10000 枚，鋳巣なしを 10000 枚ランダムに選択して読み込み，学習データとし，Mask R-CNN モデルを構築し，2 つのモデルを学習した．モデル 1 は，鋳巣を検出したいため，学習データは鋳巣のラベルのみとする．モデル 2 は，鋳巣がホイールの中に存在するため，鋳巣とホイール両方のラベルが学習データに入っている．詳細の実験条件は表 4.2 に表す．重みの初期値は Xavier 法 [23] で行った．

表 4.2　同一ホイール種の実験の条件．

	モデル 1	モデル 2
教師データ	鋳巣のみ	鋳巣とホイール
クラス数 N	2	3
学習データ数	ランダムに 20000 枚	
鋳巣あり:なし	1:1	
エポック数	1000	
学習ステップ／エポック	100	
バッチサイズ	16	
ステップサイズ	0.001	
最適化方法	Adam[24]	

4.3.2　結果および考察

2 つのモデルを用い，BKK ホイールのテスト用のフル画像 60 枚にテストを行った．結果は図 4.4，結果例は図 4.5，図 4.6 に示す．

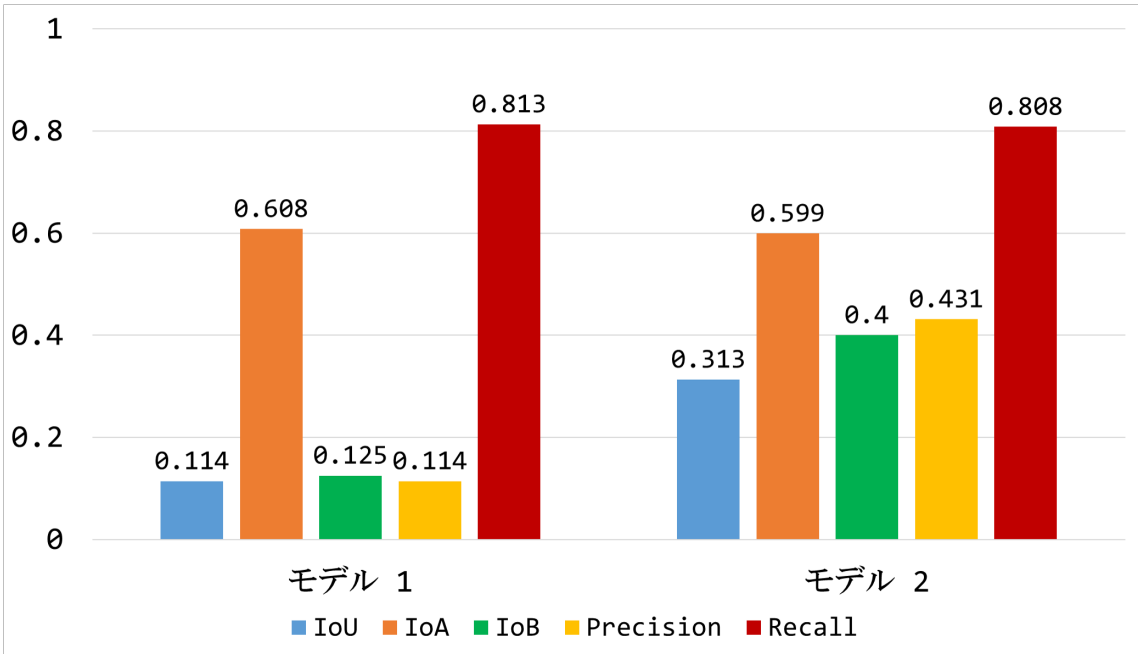
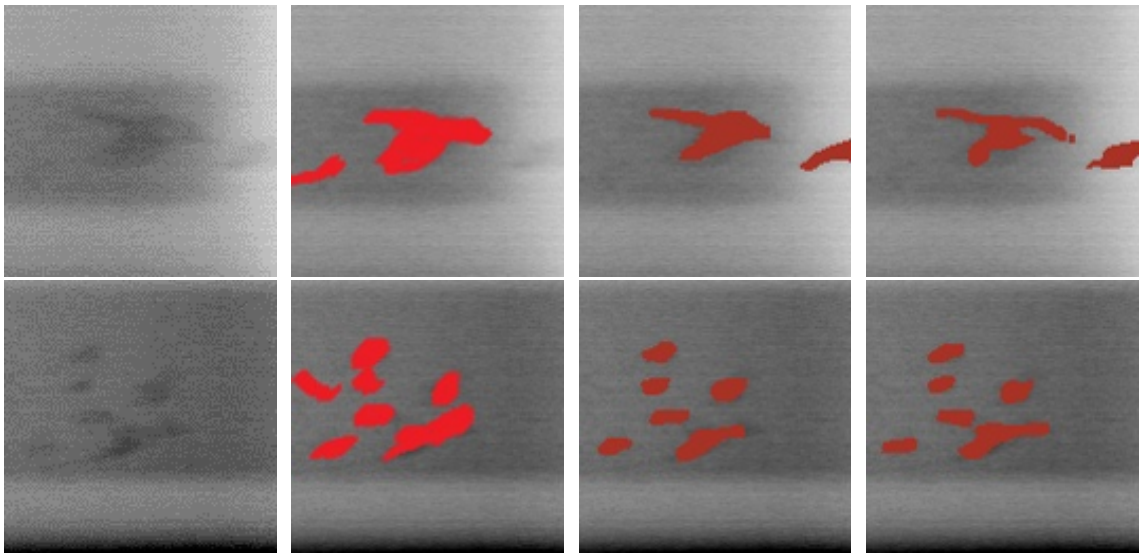


図 4.4　BKK ホイールの結果．

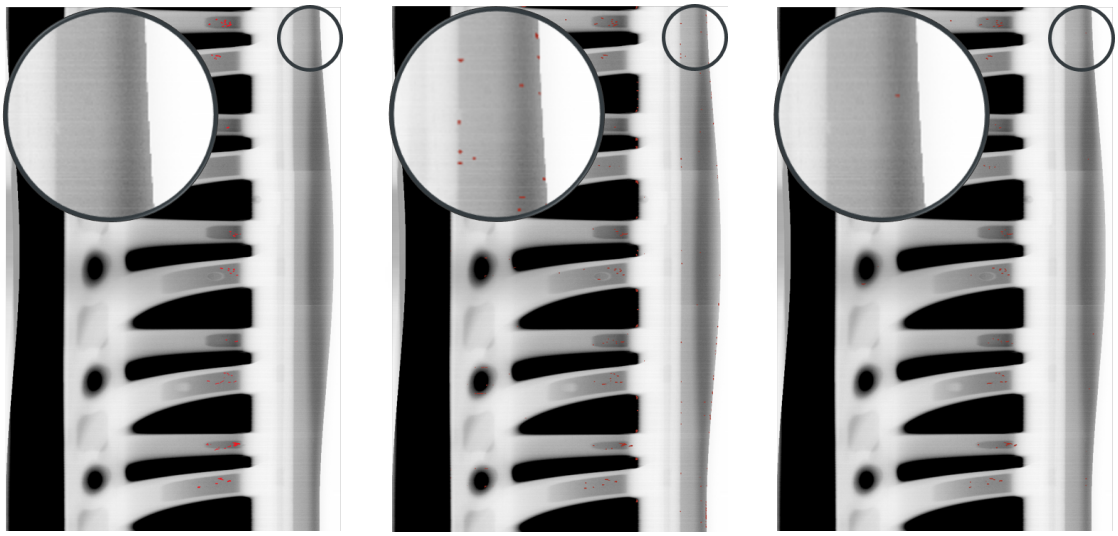
図 4.4 から，IoU について，モデル 1 は 0.114，モデル 2 は 0.313 であり，IoA について，モデル 1 は 0.608，モデル 2 は 0.599 の結果が得られた．また，IoB では，モデル 1 は 0.125，モデル 2 は 0.4 であり，Precision と Recall はモデル 1 とモデル 2 それぞれ 0.114，0.813 と 0.431，0.808 が得られた．モデル 2 に比べ，モデル 1 は IoA と Recall がわずかに良いが，IoU，IoB，Precision の方はモデル 2 が圧倒的に優れていることが確認できる．

モデル 1 とモデル 2 の Recall はそれぞれ 0.813 と 0.808 である．つまり，人間が目視で確認した鋳巣が 80% 検出できたといえる．それに，図 4.5，図 4.6 を見ると，ラベルに対していずれのモデルでも鋳巣箇所に反応しているため，人間の視覚のレベルで鋳巣検出ができると考えられる．そのため，Mask R-CNN を用いることにより自動車ホイールの X 線画像から鋳巣を検出できる可能性があると確認できる．また，図 4.6 の拡大部分を見れば，モデル 1 はモデル 2 より過検出が多いことが分かる．このような過検出が大量に存在し，モデル 1 に比較してモデル 2 の方は IoU，IoB，Precision が良い．そのため，鋳巣とホイール両方を教師データとして学習した方が過検出を低減することが確認できる．



（a）入力画像　（b）ラベル　（c）モデル 1　（d）モデル 2

図 4.5　BKK ホイールの結果例（窓画像）.



（a）ラベル　（b）モデル 1　（c）モデル 2

図 4.6　BKK ホイールの結果例（フル画像）.

4.3.3　まとめ

本実験では，自動車ホイールの X 線画像から Mask R-CNN を用い，BKK ホイールのデータでモデルを学習し，BKK ホイールにテストを行った．結果より，人間のように鋳巣を検出できる可能性があると確認した．また，鋳巣のみを教師データとして与えたモデルより鋳巣とホイール両方を教師データとして与えた方が過検出を低減できることを確認した.

4.4　複数ホイール種への適用実験

4.3 節では，Mask R-CNN を用いることにより，同一のホイール種のデータで自動車ホイールの X 線画像から鋳巣検出ができることを確認した．しかし，ホイールの内部検査の自動化を目的としたため，全てのホイール種に対応できないといけない．そのため，本実験は，4.3 節において同一のホイール種のデータで学習したモデルを複数のホイール種のデータにテストを行い，学習に使用しないホイール種の鋳巣検出の可能性を確認する．

4.4.1　条件

4.3 節のモデル 2（以降，BKK モデルと呼ぶ）を使用し，AZN，AZQ，BGD，BMB，BMC ホイールにテストを行った．

4.4.2　結果および考察

BKK モデルを用い，AZN，AZQ，BGD，BMB，BMC ホイールのデータにテストを行った結果は図 4.7 に示す．また，比較のために学習で使用した BKK ホイールの結果も含んでいる．

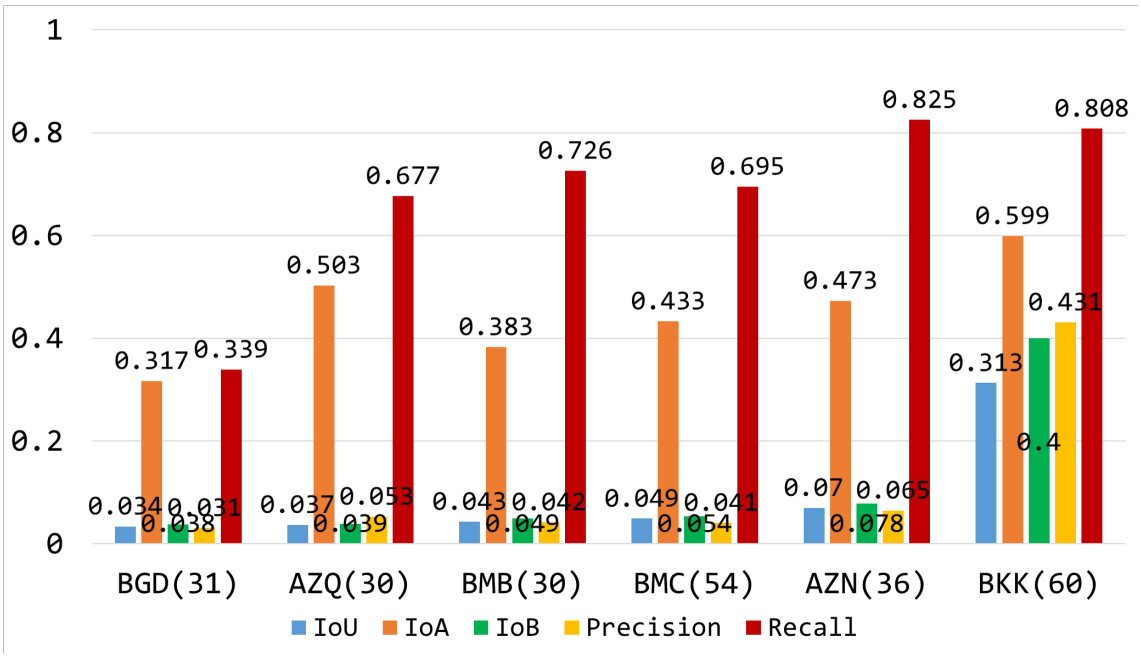


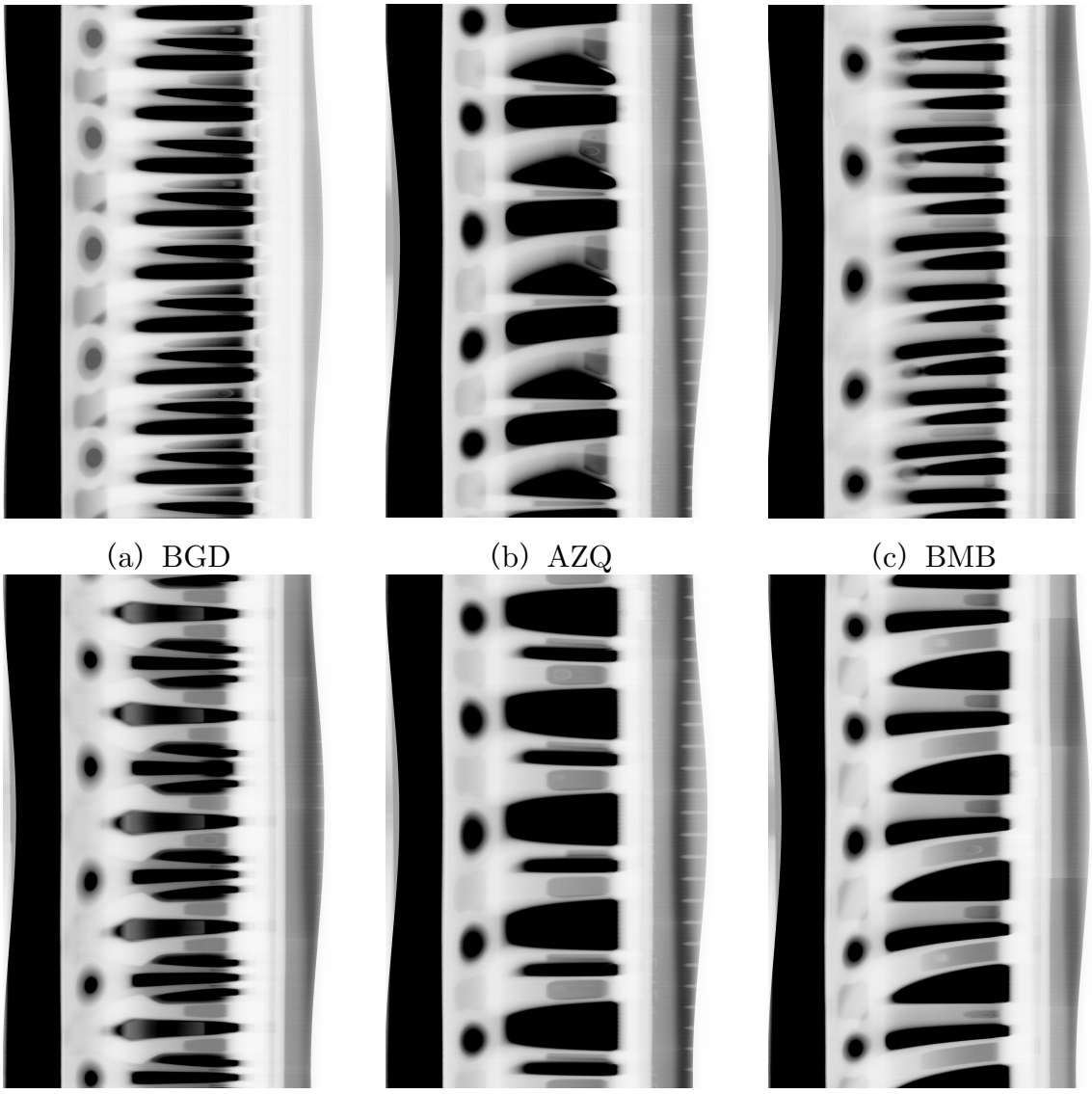
図 4.7　他のホイール種の結果．横軸はホイール名となり，括弧内の数字はテストで使用するフル画像の枚数である．

図 4.7 から BKK ホイールの結果に比べ，他のホイール種の精度は良くはない．しかし，BGD ホイール以外では，IoA と Recall の値はある程度低い値ではないことが確認できる．特に，AZN ホイールの Recall は BKK ホイールより高い結果が得られた．そのため，BKK ホイールで学習したモデルでも他のホイール種の鋳巣に対してある程度検出できることが考えられる．IoU，IoB と Precision が低い理由としては，過検出が多いことと，そのホイール特有の鋳巣が検出できないことが考えられる．また，テストで使用するホイール種の形状と BKK ホイールの形状との関係により結果が異なることも考えられる．これについては，図 4.8 を見たら，BKK ホイールの形状に比べ，BGD ホイールの形状は他のホイール種より複雑なデザインを持っており，暗い部分が多く，鋳巣がほとんどその部分に存在するため，BGD ホイールの結果がもっとも良くないことが確認できる．

この結果より，モデルの精度はホイールの形状に依存することが確認できる．全てのホイール種への対応について，全てのホイール種のデータでモデルを作成することが考えられるが，ホイールは様々な種類があり，そして毎年の新型のホイールが開発されるため，データが膨大になってしまう．そのため，各ホイール種専用モデルを作成した方が良いと考えられる．生産ライン上では，ある時間帯でどのホイール種に対して検査を行うことは決められるため，各ホイール種専用モデルを使用することは問題ないと考えられる．

4.4.3　まとめ

本実験では，BKK モデルを用い，BKK も含む複数のホイール種にテストを行った．結果より，BKK モデルは他のホイール種の鋳巣がある程度検出できるが，精度はテストで使用するホイール種の形状に依存することが確認できた．



(a) BGD　(b) AZQ　(c) BMB

(d) BMC　(e) AZN　(f) BKK

図 4.8　ホイール種の形状.（BKK モデルでのテスト結果順）

4.5　転移学習実験

4.4 節では，モデルの精度がテストで使用するホイール種の形状に依存することを確認した．全てのホイール種に対応させるためには，各ホイール種専用モデルの作成が必要である．しかし，学習に必要なラベルの生成には人間の手間が不可欠であり，ホイール種類は毎年どんどん増えていくため，大量の学習データを作成することは非常に困難となっている．そのため，生産ライン上で自動検査と目視検査を並行してデータを蓄積することが現実的であると考えられる．しかし，自動検査のため，各ホイール種に対してある程度精度が良いモデルも必要であるため，本実験は，ホイールのデータ不足の対策とし，BKK モデルをベースにして転移学習を用いることにより，AZN と BMC ホイール種に対して少ないデータ数でもデータが多いモデルと同等もしくはそれ以上の精度が取得できることを確認する実験である．

4.5.1　条件

AZN と BMC ホイールのデータを用い，BKK モデルをベースとし，転移学習を行った．しかし，BKK モデルは他のホイール種の鋳巣は反応しているため，BKK モデルのヘッドネットワークはある程度仕事ができると考えられる．そのため，AZN と BMC ホイールに対する転移学習モデルは 2 つの場合を行った．1 つ目は，BKK モデルの全部の重みを初期値とし，専用ホイール（AZN あるいは BMC）のデータでヘッドネットワークを含む全ての重みを更新する（以降，ヘッド有と呼ぶ）．2 つ目は，BKK モデルの全部の重みを初期値とし，専用ホイールのデータでバックボーンネットワークと RPN の重みのみを更新する（以降，ヘッド無と呼ぶ）．また，比較対象として専用ホイールを用いて BKK モデルと同じ条件で学習したモデルも作成する（以降，専用モデルと呼ぶ）．ただし，学習データには鋳巣ありと鋳巣なしの枚数が半分ずつとし，専用モデルのデータ数は足りないため 6000 枚にした．そして，BKK モデル，専用モデル，ヘッド有モデルおよびヘッド無モデル合計 4 つのモデルの比較を行った．詳細な条件は表 4.3 に示す．残りの条件は 4.3 節の実験と同じように設定した．AZN と BMC を選択したのはデータ数がもっとも多いためである．

表 4.3　複数のホイール種への対応の実験

	BKK モデル	専用モデル	ヘッド有	ヘッド無
初期値	Xavier 法		BKK モデルの全部の重み	
学習部分	全部の重み		ヘッドネットワークを含む全ての重み	バックボーンネットワークと RPN の重み
ホイール種	BKK	専用ホイール		
学習データ数	ランダムに 20000 枚	ランダムに 6000 枚	ランダムに 1000 枚	
エポック数	1000		500	
教師データ	鋳巣とホイール			

4.5.2　結果および考察

BKK モデル，専用モデルそして 2 つの転移学習モデルを用い，AZN，BMC のテスト用のフル画像それぞれ 36，54 枚にテストを行った．結果は図 4.9，図 4.10 に示す．

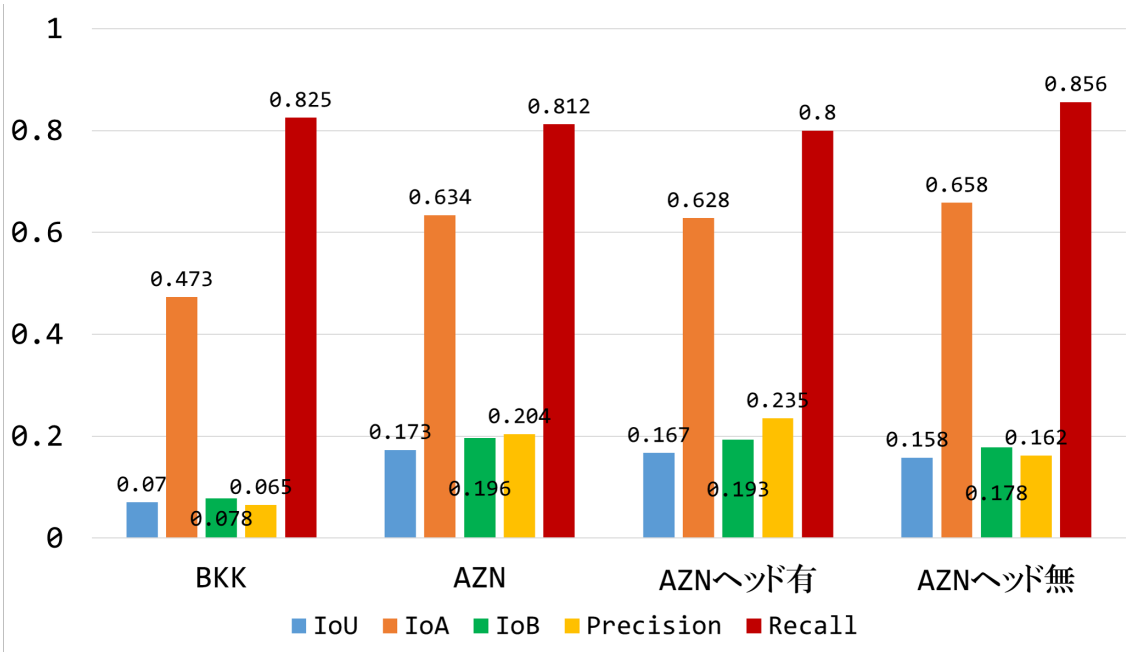


図 4.9　AZN ホイールの結果.

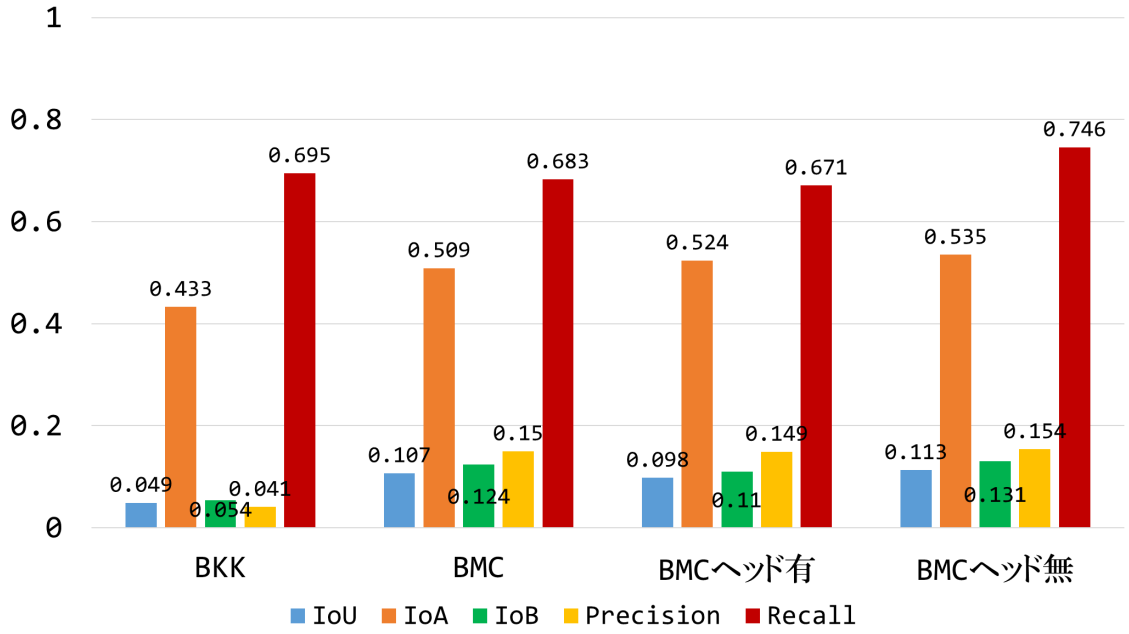
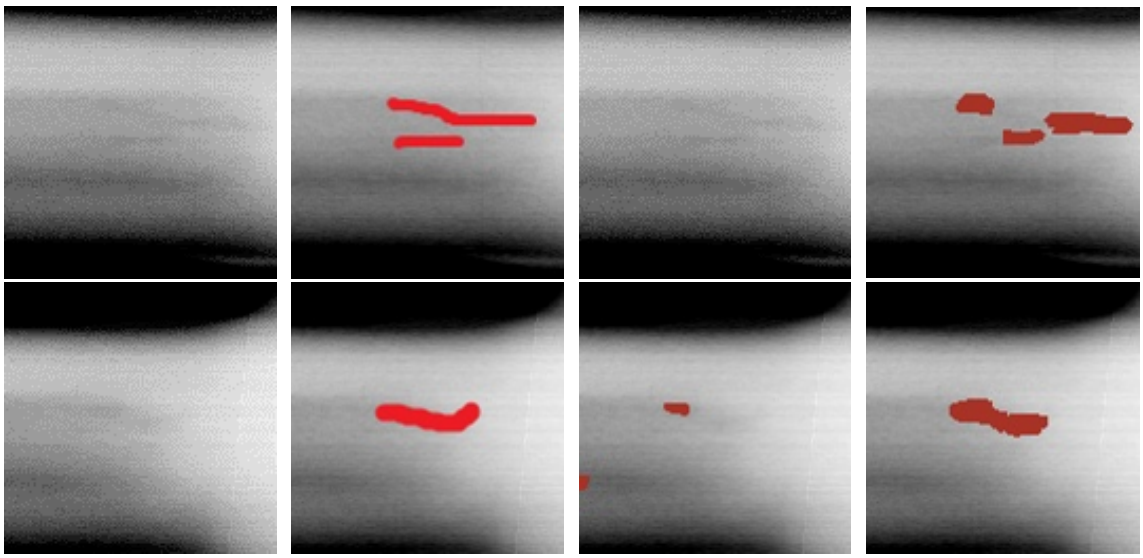


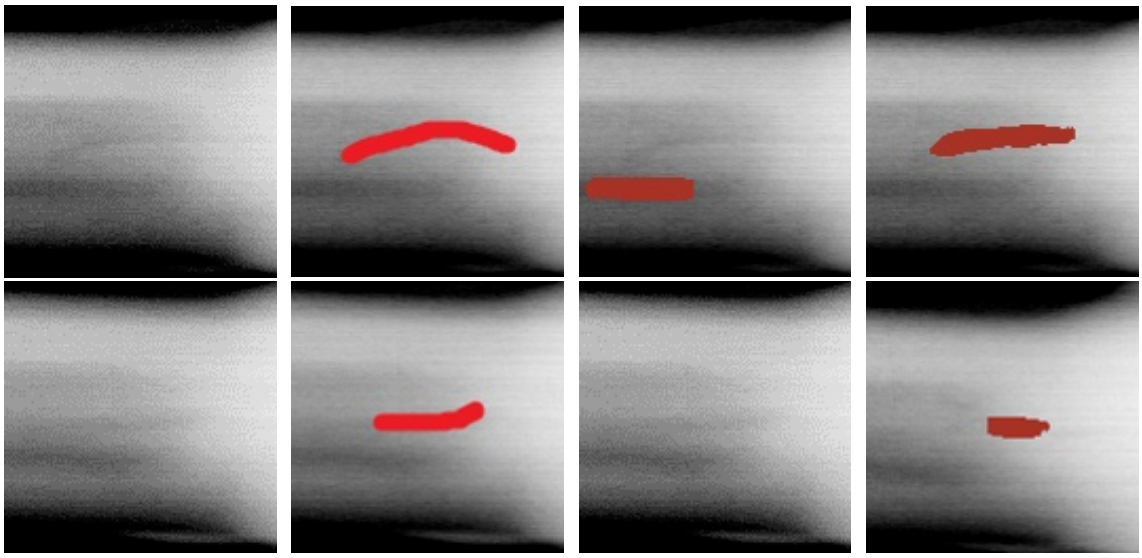
図 4.10　BMC ホイールの結果.

図 4.9，図 4.10 の結果から，いずれの転移学習モデルでも BKK モデルより全ての評価指標の結果が良くなった精度が得られた．しかし，Recall については変化が多くはないため，転移学習を用いることより主に過検出が低減すると考えられる．そして，専用モデルに比べていずれの転移学習モデルでも同等もしくはそれ以上の精度が得られ，転移学習は効果があると確認できる．結果例は図 4.11，図 4.12，図 4.13，図 4.14 に示す．



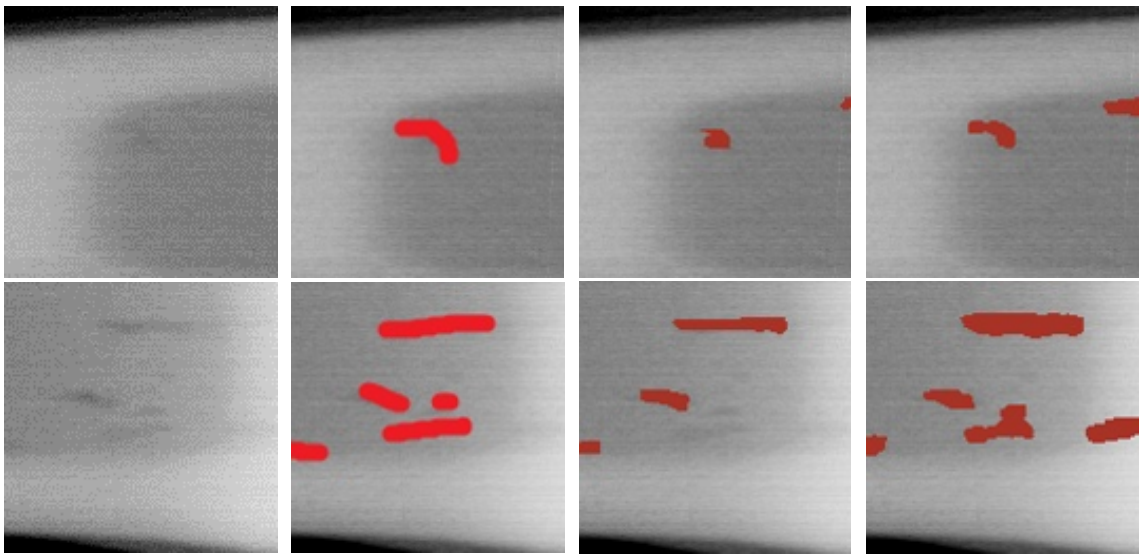
(a) 入力画像　(b) ラベル　(c) BKK モデル　(d) AZN ヘッド有

図 4.11　AZN ホイールのヘッド有の結果例（窓画像）.



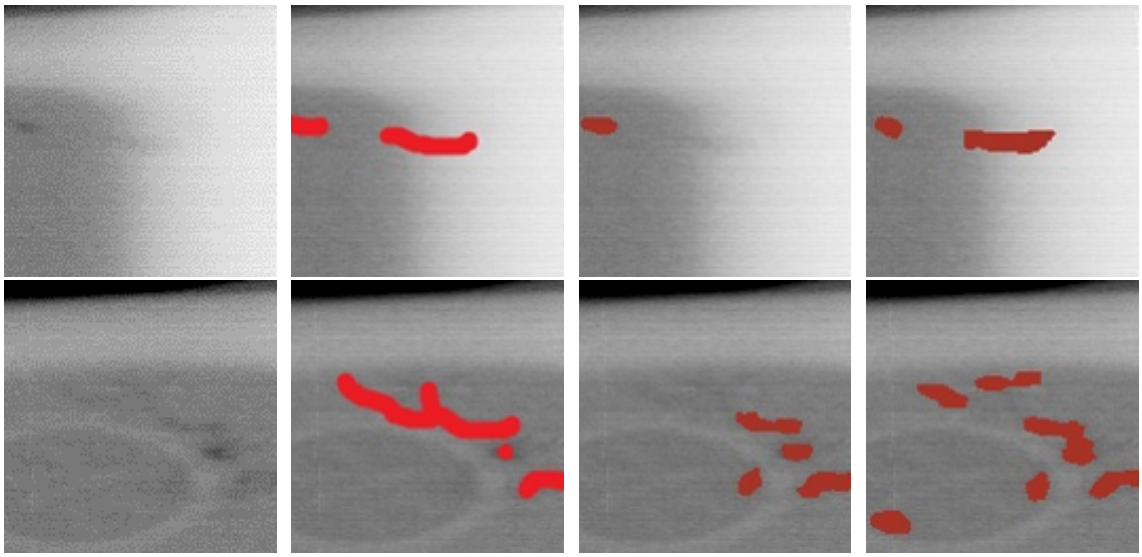
(a) 入力画像　(b) ラベル　(c) BKK モデル　(d) AZN ヘッド無

図 4.12　AZN ホイールのヘッド無の結果例（窓画像）.



(a) 入力画像　(b) ラベル　(c) BKK モデル　(d) BMC ヘッド有

図 4.13　BMC ホイールのヘッド有の結果例（窓画像）.



（a）入力画像　（b）ラベル　（c）BKK モデル　（d）BMC ヘッド無

図 4.14　BMC ホイールのヘッド無の結果例（窓画像）.

ヘッド有とヘッド無の結果を比較すると，AZN ホイールの場合では，IoU，IoB，Precision はヘッド有の方が良いが，IoA と Recall はヘッド無の方が優れている．一方, BMC ホイールの場合ではヘッド無の方が完全に優れている．これについて, BKK モデルは他のホイールに対して過検出が多いことと，そのホイール特有の鋳巣が検出できないため，バックボーンネットワークと RPN は学習すべき部分となる．ヘッド無はそれらの部分を主に更新し，特有の鋳巣がヘッド有より検出できるようになったため，ヘッド有より良い IoA と Recall が得られた．また，転移学習で使用するホイール種とベースにしたホイール種との関係であると考えられる．図 4.7 の結果では AZN の結果の方は BMC の結果よりどの評価指標の値も高く，図 4.8 の形状から，AZN ホイールは BMC ホイールより明度など BKK ホイールと似ている形状を持っていると考えられる．BMC ホイールは，BKK ホイールと形状が大きく異なるため，特有の鋳巣が多く検出でき，ヘッド有のモデルが学習不足であり，ヘッド無の方が良いと考えられる．一方，AZN ホイールの場合，ヘッド無の方は特有の鋳巣が検出できるが，BKK ホイールに近い形状を持っているため，ヘッドネットワークを学習することで鋳巣がより正しく検出でき，IoU，IoB，Precision についてはヘッド有の方が良いと考えられる．しかし，実際には検出できない方より過検出の方が良いため，全体的にヘッド無の方が優れていると考えられる．

4.5.3　まとめ

本実験では，データ不足の対策とし，各ホイール種に対して良い精度のモデルを作成するため，転移学習を用いることを提案した．BKK モデルをベースにして AZN，BMC ホイールでヘッド有とヘッド無 2 つの転移学習モデルを学習した．結果より，いずれのホイールと転移学習モデルでも，専用ホイールと同等もしくはそれ以上の精度が得られた．そのため，転移学習を用いることにより少ないデータ数でも効果があることが確認できた．また，現在の条件では，転移学習においてヘッド有よりヘッド無の方が良いと確認できた．

4.6　転移学習のデータ数の実験

4.5 節では，転移学習により，データが少ないホイール種でも専用モデルの作成ができることを確認した．本実験では，転移学習で必要となるデータ数を検討する．

4.6.1　条件

4.5 節と同じ条件で，AZN と BMC ホイールを用い，BKK モデルをベースにしてヘッド無の転移学習モデルのみを作成した．データ数は 100 から 2000 まで（鋳巣ありと鋳巣なしの枚数は半分ずつ）変更し，BKK モデルおよび専用モデルと比較を行った．

4.6.2　結果および考察

AZN，BMC のテスト用のフル画像それぞれ 36，54 枚にテストを行った．結果は図 4.15 と図 4.16 に示す．結果より，いずれのホイール種でも，データ数 100 枚の場合は学習データが少なくて精度があまりないことが確認できる．また，AZN ホイールの場合では，データ数 200 枚と 300 枚のモデルは IoU，IoA，IoB の値が専用モデルに近いが，Precision と Recall の値が小さい．データ数 400 枚のモデルは Precision と Recall が専用モデルにほぼ同じ値であるが，IoU，IoA，IoB の値が良くない．データ数 500 枚以降は専用モデルと同等であることが考えられる．一方，BMC ホイールの場合では，専用モデルに比べ，データ数 200 枚のモデルは IoU，IoA，IoB が良くない．500 枚と 2000 枚のモデルも IoU，IoA，IoB の値がわずかに小さいが，Precision と Recall は専用モデルより優れている．残りのモデルは専用モデルと同等もしくはそれ以上の精度である．

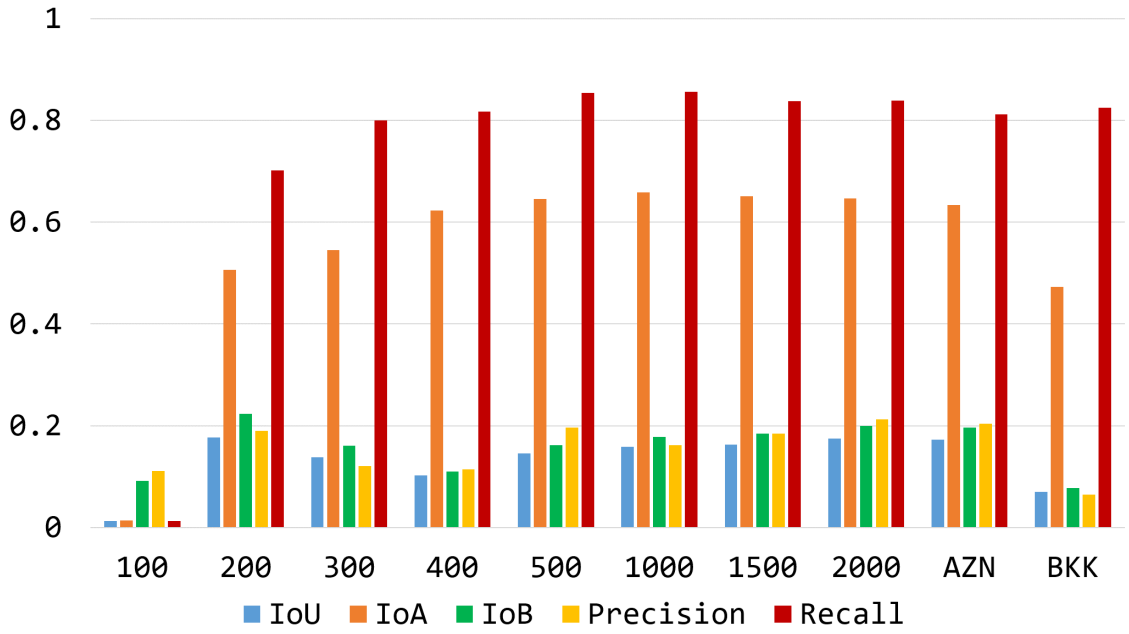


図 4.15　AZN ホイールの転移学習データ数の結果．横軸は AZN の転移学習モデルと最後 2 つは AZN 専用モデルと BKK モデルである．

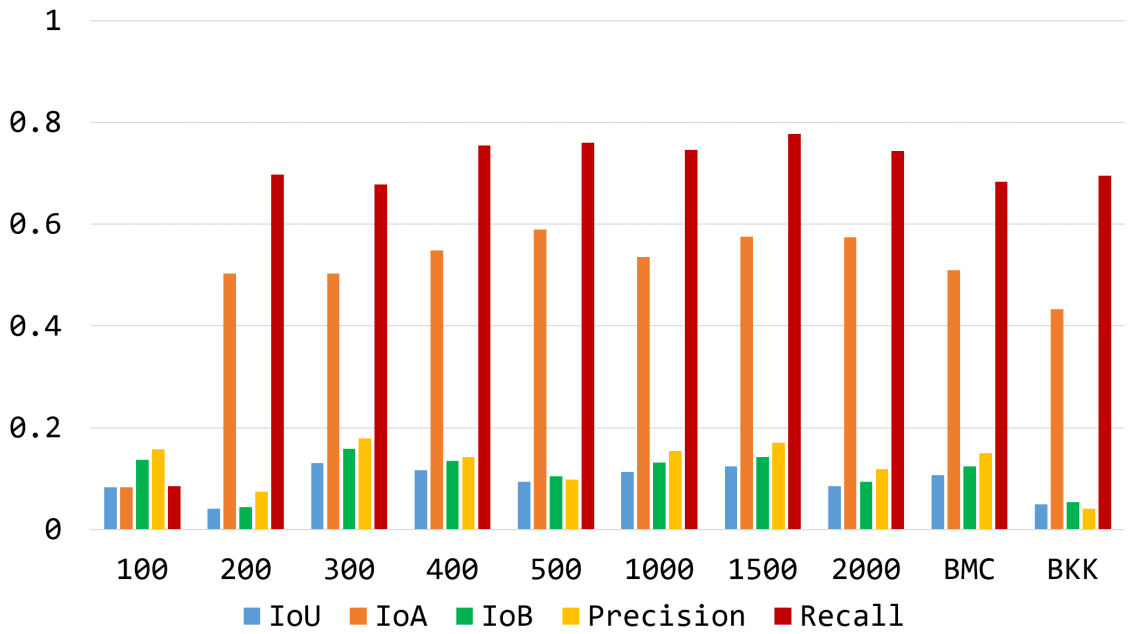


図 4.16　BMC ホイールの転移学習データ数の結果．横軸は BMC の転移学習モデルと最後 2 つは BMC 専用モデルと BKK モデルである．

上記の結果より，転移学習の効果は再度確認でき，学習データ数は 1000 以下でも専用モデルと同等もしくはそれ以上の精度が得られるが，使用したホイール種の形状とベースにしたホイール種の形状の関係に依存するため，必要となるデータ数の最低限はまだ不明であり，今後検討する必要がある．

4.6.3　まとめ

本実験では，転移学習で必要となりデータ数を検討した．結果より，転移学習の効果は再度確認でき，学習データ数は 1000 以下でも良い精度が得られることを確認した．

第 5 章

おわりに

本論文では，自動車ホイール内に存在する鋳巣と呼ばれる結果の自動検出を目的とし，人間が確認した鋳巣のデータを用い，Mask R-CNN を適用し，自動車ホイールの X 線画像から高精度で鋳巣を検出することを目指した．また，様々なホイール種への対応について，各ホイール種専用モデルの作成を解決策とし，データ不足の対策として転移学習を用いることを提案し，転移学習により少ないデータ数でもデータ数が多い場合と同等もしくはそれ以上の精度を目指した．

第 1 章では，自動車ホイールの鋳巣検査について述べ，本論文の研究目的について述べた．第 2 章では，本論文で用いた基礎的知識である畳み込みニューラルネットワークと Mask R-CNN について説明した．第 3 章では，実験で用いたホイールの X 線画像データについて説明し，データ切り出しおよびラベリングの方法について述べた．第 4 章では，本研究で行った実験について述べた．具体的に，用いた評価方法を説明し，同一のホイール種の適用の実験では，鋳巣のみを教師データとして与えたモデルと，鋳巣とホイール両方を教師データとして与えたモデルの 2 つを比較した．鋳巣とホイールのモデルは過検出が低減し，鋳巣のみのモデルより良い結果が得られ，Mask R-CNN の鋳巣の検出可能性を確認した．また，複数のホイール種への適用の実験では，鋳巣とホイールのモデルを用い，複数のホイール種にテストを行った．モデルの精度はテストで使用するホイール種の形状に依存することが確認できた．さらに，転移学習の実験では，鋳巣とホイールのモデルをベースとし，他 2 つのホイール種を用い，ヘッド有とヘッド無 2 つの転移学習モデルを作成し，ベースモデル，専用モデルと比較を行った．いずれの転移学習モデルでも，ベースにしたモデルより良い精度が得られ，専用モデルと同等もしくはそれ以上の精度が得られた．それに，ヘッド有に比べ，ヘッド無の方が優れていることが確認できた．

謝辞

本研究を進めるにあたり，日頃より丁寧なご指導ご鞭撻を頂きました，本学電気電子情報工学専攻 杉田 泰則 准教授に深謝いたします．また，本論文の審査において貴重なご意見を賜りました，本学電気電子情報工学専攻 岩橋 政宏 教授，並びに本学電気電子情報工学専攻 圓道 知博 准教授に御礼申し上げます．さらに，データ提供や学習用ラベルの作成，有意義な議論を下さった，株式会社明和ｅテック開発部の皆様に感謝いたします．そして，本研究を進めるにあたり，多くのご指摘ご助力を下さった，杉田研究室の同期・後輩の皆様に感謝の意を表します．最後に，学生生活を常に支え続けてくれた両親・家族に心から感謝いたします．

令和２年２月７日

参考文献

[1] 日本国特許庁，公開特許公報，トヨタ自動車株式会社，“鋳巣計測方法，”特開 2005-351875，2005 年 12 月 22 日公開.

[2] 日本国特許庁，公開特許公報，日立 GE ニュークリア・エナジー株式会社，“超音波探傷装置及び超音波探傷方法，”特開 2012-27037，2012 年 02 月 09 日公開.

[3] 日本国特許庁，公開特許公報．日本精工株式会社，“巣の評価方法，”特開 2018-179950，2018 年 11 月 15 日公開.

[4] 日本国特許庁，公開特許公報，株式会社日立製作所，“鋳造品の検査方法および検査装置，”特開 2019-143992，2019 年 08 月 29 日公開.

[5] 日本国特許庁，公開特許公報，株式会社明和 e テック，“鋳造品の鋳巣検出装置および鋳巣検出方法，”特開 2018-96955，2018 年 06 月 21 日公開.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, “Mask R-CNN,” *IEEE International Conference on Computer Vision*, pp.2980-2988, 2017.

[7] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems 25*, pp.1097-1105, 2012.

[8] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *International Conference on Learning Representations*, 2015.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-9, 2015.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun “Deep Residual Learning for Image Recognition,” *The IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision* **115**, pp.211-252, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp.580-587, 2015.

[13] Ross Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision*, pp.1440-1448, 2015

[14] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems 28*, pp.91-99, 2015.

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2016.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision*, pp.21-37, 2016.

[17] Xinlei Chen, Ross Girshick, Kaiming He, Piotr Dollár, "TensorMask: A Foundation for Dense Object Segmentation," *Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Yann LeCun, Yoshua Bengio, Geoffrey E. Hinton, "Deep learning," *Nature* **521**, pp.436–444, 2015.

[19] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, "Feature Pyramid Networks for Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp.936-944, 2017.

[20] 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," *European Conference on Computer Vision*, pp.740-755 2014.

[21] waspinator, “pycococreator,” *GitHub*.
`https://github.com/waspinator/pycococreator`.
December 5th 2018.

[22] Waleed Abdulla, “Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow,” *GitHub*.
`https://github.com/matterport/Mask_RCNN`.
November 6th 2017.

[23] Xavier Glorot, Yoshua Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp.249-256. 2010.

[24] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations*, 2015.

発表実績

- グエンティエンアイン，杉田泰則，“深層学習を用いた自動車ホイールの鋳巣検出に関する一検討，”電気学会制御研究会，CT-19-112，pp.39-44，July 2019.
- グエンティエンアイン，杉田泰則，“自動車ホイールの欠陥検出に対するインスタンスセグメンテーション手法の適用，”電子情報通信学会 信号処理研究会 信学技報，vol.119，no.185，SIP2019-48，pp.45-50，August 2019.

付録

A　ホイールフル画像の撮影

自動車ホイールの X 線画像の撮影は図 A.1 のように，ホイールを撮影機にあるカーボンマットに置き，その上にラインセンサーがある．そのマットが回転しながら，ラインセンサーがそれに対するホイールの部分を撮影し，マットが 1 回転したら，ホイール全体が撮影できる．ラインセンサーに対する全てのホイールの部分を結合すると，ホイールフル画像が得られる．そのため，自動車ホイールは円形であるが，撮影したホイールの X 線画像は長方形になる．

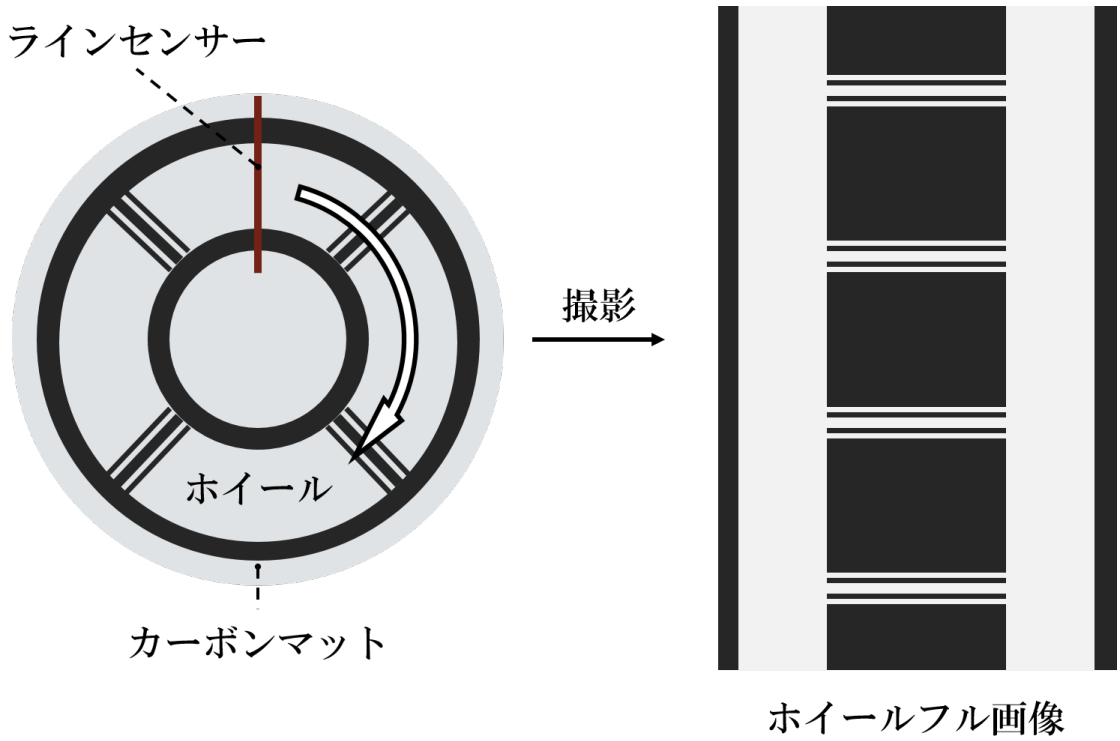


図 A.1　ホイールフル画像の撮影.

B　学習データ数を変更した BKK ホイールの結果

4.3 節では，BKK ホイールのデータ画像 20000 枚を用い，BKK モデルを作成した．本章では，学習データ数を変更し，BKK ホイールにテストを行って比較した．結果は図 B.2 に示す．

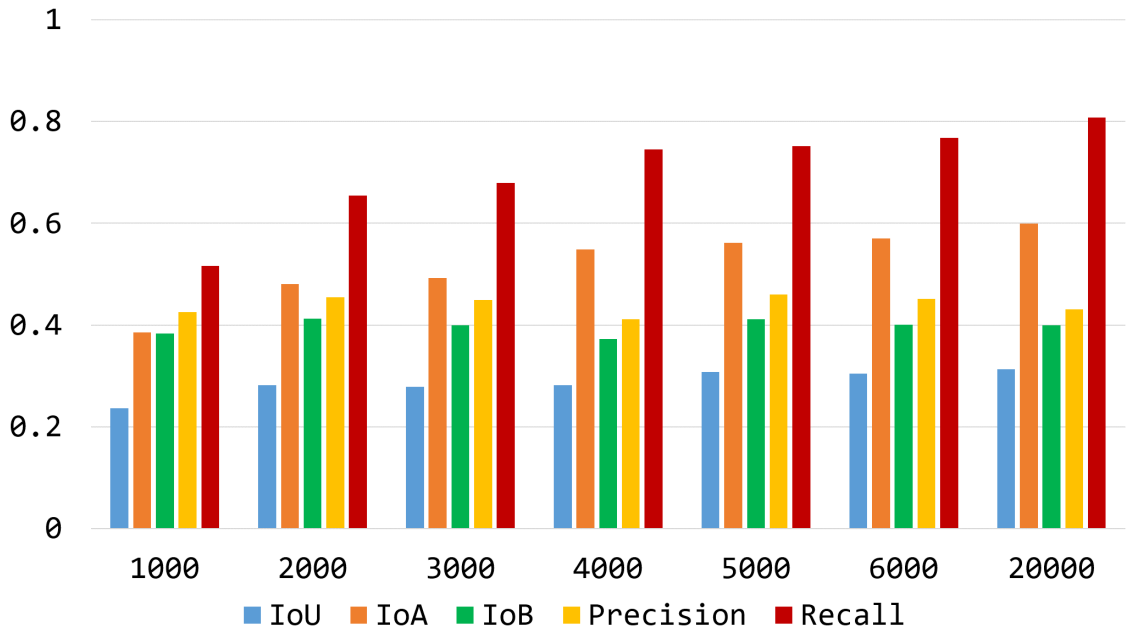


図 B.2　学習データ数を変更した BKK ホイールの結果．

図 B.2 より，学習データ数を増やすと，Precision と Recall の値が大幅に増加することが確認できた．また，学習データ数が増えたが，IoU の精度が少しだけ良くなり，IoA と IoB については学習データ数との相関関係は見られなかった．その理由については，学習時に学習時間を制限するためにエポックごとの学習ステップを 100 にしたため，学習データ数が増えても，エポックごとの使用データ数は同じであり，精度にあまり差がないと考えられる．さらに，AZN モデルおよび BMC モデルと比べ，同じ学習データ数でも BKK ホイールの方は精度が良いことが確認できた．これについては BKK ホイールは他のホイール種より形状が複雑でないため，学習しやすいと考えられる．

C　閾値 T を変更した BKK ホイールの結果

本論文では，Precision と Recall の評価において，正解鋳巣の閾値 $T = 0.2$ にした．本章では，BKK モデルを用い，閾値 T を変更して BKK ホイールのデータにテストを行った．結果は図 C.3 に示す．

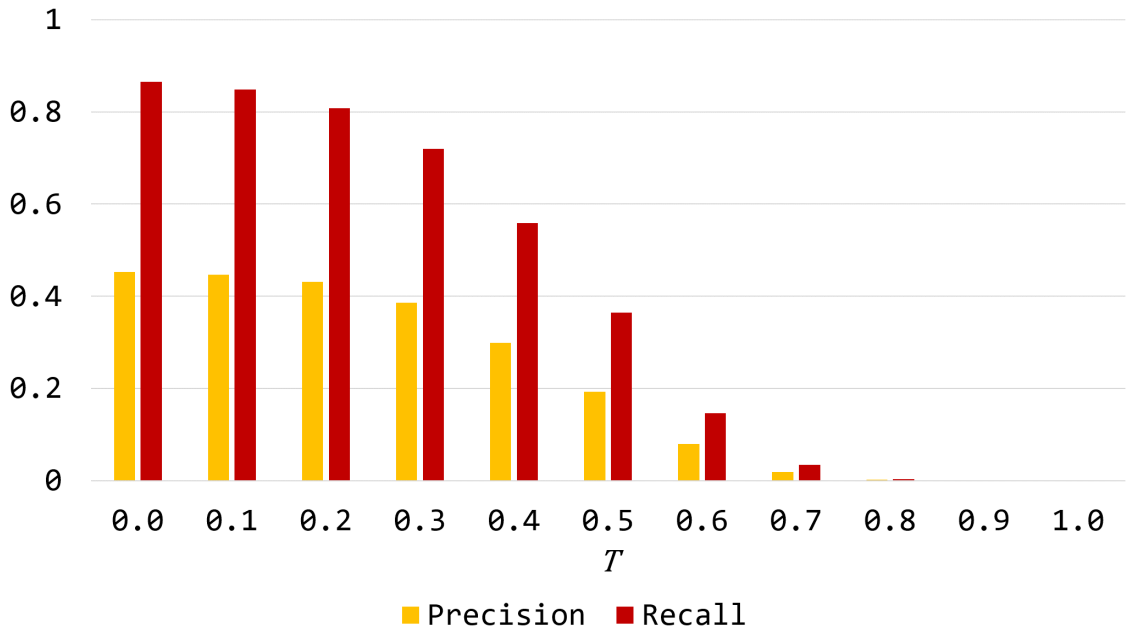


図 C.3　閾値 T を変更した BKK ホイールの結果．

図 C.3 より，閾値 T が大きくなると，Precision と Recall の値が小さくなり，最後 0 となることが確認できる．また，閾値 T が 0 ～ 0.3 の範囲は Precision と Recall の差があるが，0.3 ～ 0.6 の範囲に比べ，0.3 ～ 0.6 の方は Precision と Recall の値が急激に減少しているため，ラベルとの IoU が 0.3 ～ 0.6 の鋳巣の割合が最も多いことも考えられる．

D　BGD ホイールの転移学習結果

4.5 節では，転移学習により，AZN と BMC ホイールを用い，少ないデータで専用モデルの作成ができることを確認した．しかし，4.4 節の結果において，AZN と BMC ホイールはある程度検出精度が良いホイール種であり，BGD など精度が良くないホイール種の場合なら転移学習は効果があるかどうか不明である．そのため，4.5 節の転移学習実験と同じ条件で BGD ホイール種を用いて転移学習モデルを作成した．

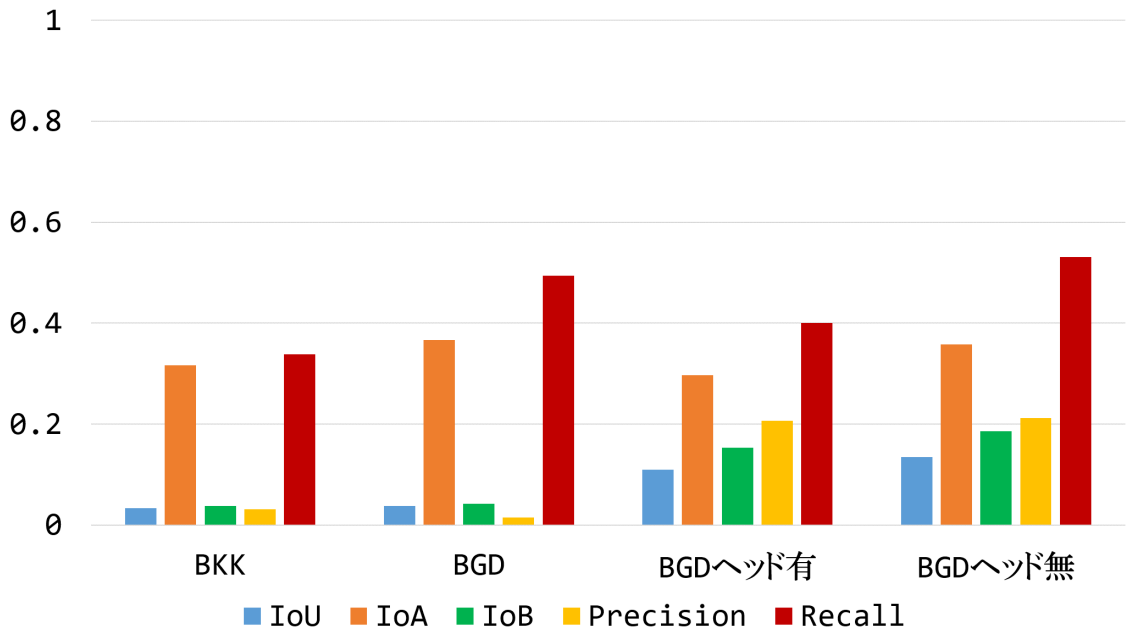


図 D.4　BGD ホイールの結果．BGD 専用モデルの学習データは足りないため 4600 であり，テスト枚数は 31 枚である．

図 D.4 の結果から，BGD ホイールの場合でも，AZN と BMC ホイールのように，いずれの転移学種モデルでも BKK モデルより精度が良く，専用モデル以上の精度が得られたことが確認できた．また，ヘッド無の方が優れていることも分かった．

E　テスト時間

フル画像の切り出しのように，テストを行う際も，移動幅 $h_c/2$ と $w_c/2$ を考慮してサイズ $h_c \times w_c$ のスライド窓を移動させながら，その窓に対してテストを行う．しかし，窓に対して 1 枚ずつテストするには，非常に時間がかかる．そのため，スライド窓を移動させながら，M 個の窓を重ねて同時にテストを行うことが考えられる．BKK モデルを用い，M を変更して BKK ホイールのテスト用 60 枚のフル画像にテストを行い，テスト時間を測定した．結果は図 E.5 に示し，$M = 14$ の場合はテスト時間が最も短い（35.275s）と確認できた．

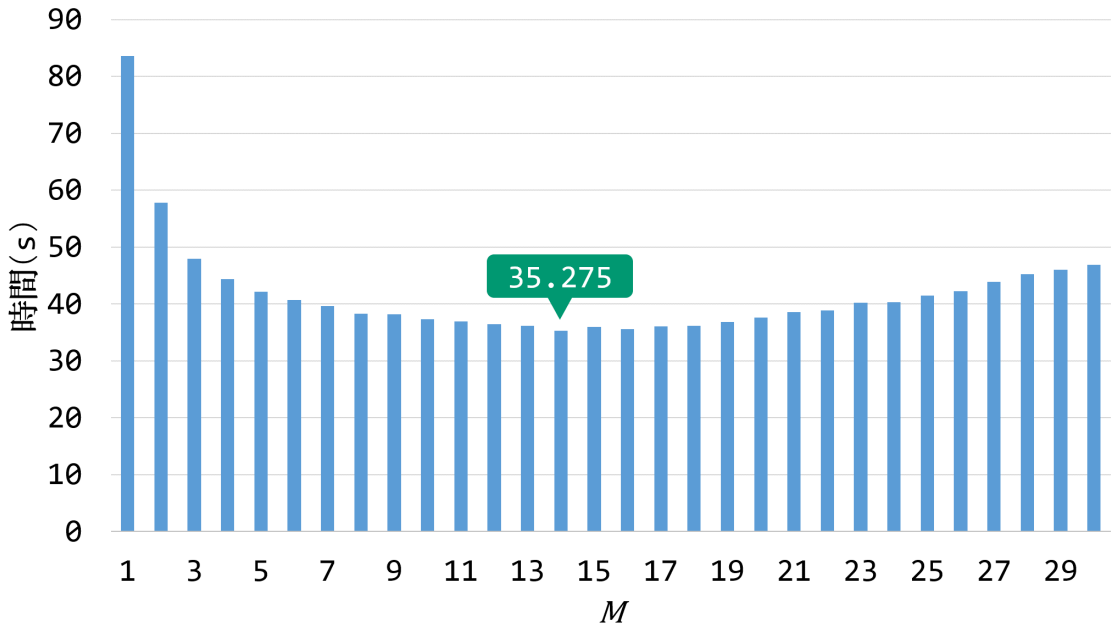


図 E.5　M 個の窓を重ねて同時にテストを行った結果．メモリ不足のために M が 31 以上の場合は実行できない．

F 実験環境

本論文での実験環境は表 F.1 に示す.

表 F.1 実験環境.

OS	Windows 10 Pro
CPU	Intel® Core™ i5-6600
メモリ	32GB
GPU	NVIDIA Geforce GTX 1070 8GB
プログラミング言語	Python
TensorFlow	1.9.0
Keras	2.2.4