修士論文

CNN を用いた座標情報を利用しない

物体位置予測手法の提案及び免疫細胞画像における検証

The proposal of the CNN without the coordinate information

for predicting the object location

and its validation on Immune cell images

報 告 者

学籍番号: 1245054

氏名: 楠瀬 翔也

指 導 教 員

星野 孝総 准教授

令和 4 年 2 月 18 日

高知工科大学大学院工学研究科

基盤工学専攻電子・光工学コース

目 次

第1章　序論

1.1.　研究背景

近年，ハードウェアの計算能力の向上によって大規模な並列計算を必要とする深層学習が広く用いられるようになった．深層学習は人工知能技術を支える大きな技術の一つで，画像処理，音声処理，自然言語処理などの様々な分野で応用されている [1–6]．特に画像処理の分野においては様々なタスクに特化した手法が研究されており，物体認識，物体検出，セマンティックセグメンテーション，画像生成など多岐にわたる [7–13]．また，大規模な計算モデルを用いた高精度な処理や，エッジデバイスでの運用のために考えられた軽量なアーキテクチャなど，様々なユースケースを対象とした研究も行われている [14–17]．画像処理における深層学習は，畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) によって様々なタスクで大きな成果を示している．これは，2012 年に開催された大規模画像データセットを用いた画像認識コンペティションの Large Scale Visual Recognition Challenge(ILSVRC) [18] において，Krizhevsky 氏によって発表された CNN モデルの AlexNet が優勝したことが背景にある [19]．同コンペティションは，前年度まではハンドクラフト特徴量を用いたサポートベクトルマシン (support-vector machine: SVM) による認識技術が優秀な成績を収めていたが，2012 年の優勝手法である AlexNet を筆頭に，2013 年は ZFNet，2014 年は GoogLeNet と CNN モデルの優勝が続き，CNN モデルが上位を占めるようになった [20–24]．この活躍は研究分野のみならず，商業分野などにも用いられるようになり，医療現場や工場生産自動化 (Factory Automation: FA) の現場での実用化が進められている [25–27]．

しかし，このような輝かしい成果の裏には学習に必要なデータセットの作成のための労力が隠れている．基本的な CNN による物体認識や物体検出は教師あり学習であり，学習に用いる画像に対してその画像がなんなのか，画像内のどこに何が写っているかという正解ラベルを必要とする．そのため教師あり学習を行うには，用いる画像に正解ラベルを人間によって定義する必要がある．前述の ILSVRC ではすでにラベルがつけられた学習データを用いた分類を競うコンペティションだが，現実世界のタスクにおいて，撮影した写真にラベルをつける作業は現場の人間による作業となる．より正確な推論モデルを作成するにはより多くのデータを用いることが精度を確保する手段の一つだが，多くのデータを用いようとするほど多くのアノテーション作業を必要とする．特に，物体検出タスクにおいては，1 枚の画像の中に映る物体のクラスとその座標情報の両方を紐づける必要があり，複数物体が写っている場合は全てに対応する必要がある．こういった作業は，自動化システムの導入前に直面する大きな課題の一つであると

言える．

また，近年免疫学の分野において，免疫細胞に関する研究が盛んに行われている．免疫細胞とは人間の免疫機関の細胞の一部で，人間の健康を維持する上で大切な細胞の一つである．免疫細胞は体組織の中に存在しており，偽足と呼ばれる突起を用いて移動を行う遊走性細胞である．偽足を用いた移動はアメーバのような形状の変化を伴っている．免疫細胞は，医療機関などで活動解析を行う場合，顕微鏡を用いた撮影が行われ，その撮影データを元に解析されることが多い．顕微鏡画像に写る免疫細胞を図 1.1 に示す．免疫細胞の活動解析は，あらゆる病気の診断のために用いることができると考えられており，特に免疫細胞の移動速度が子宮内膜症の発見に関係しているとして研究が行われている [28,29]．この研究における免疫細胞の解析作業は現在手作業によって行われている．ここでいう解析作業とは，撮影された動画に映る複数の免疫細胞のうちいくつかを選択したのち，それらを数十フレーム間追跡することで免疫細胞の移動速度や活発度といった活動量を測定することを意味する．しかしながら，解析を必要とする画像の枚数が多く，さらに 1 枚の画像に多くの細胞が写っているため 1 回の解析作業に多くの時間がかかってしまうという問題点がある．一般的な解析システムには，免疫細胞の動きがわかる程度に撮影フレームを間引きした動画が撮影されており，後者の問題点は解決されているものの，解析作業については解析作業者に負担となっている．そのため，免疫細胞の選択や追跡，追跡結果を基にした活動量の解析を自動で行うようなコンピュータツールなどによって，医療従事者の支援ができると考えられる．以前医療従事者が選択した免疫細胞を数フレーム間にわたって追跡を行うツールが開発された [30] が，選択部分において自動化ができないという問題があった．この問題に対して機械学習を用いた物体検出手法を用いることで自動選択が可能であると考えられたが，免疫細胞の学習用データを作成するためには，医療従事者の知識と労力が必要である．また，撮影機材や拡大率，解像度，照明条件などが様々であるため，インターネットで公開されている免疫細胞のデータセットを本件に用いることは難しかった．自動選択を実現しつつ医療従事者が労力を払わないためには，位置情報を用いずに免疫細胞の位置を予測できる必要がある．

1.2. 研究目的

本研究では，検出タスクにおける座標情報のアノテーションレスな物体位置予測の実現を目的とする．本来検出タスクというのは，画像内に映る物体を正確に位置情報とクラス情報を合わせて推論するタスクのことであり，画像がどのクラスに属するかのみを推論する認識タスクとは異なるものである．そのため物体検出を教師あり学習によって学習するには，各教師データの画像に正解ラベルとしてクラス情報と座標情報を持たせる必要がある．しかしながら，座標情報は画像においてはピクセル単位の細かい数値で，アノテーション作業もクラスの振り分け以上に大変な作業である．そのため物体の座標情報無しに物体の写る位置をある程度予測できるようなアノテーションコストの低い物体位置予測手法を提案する．ここで，本稿における物体位置予測とは物体検出とは異なり，画像内に存在する全ての物体を正確に取得することを

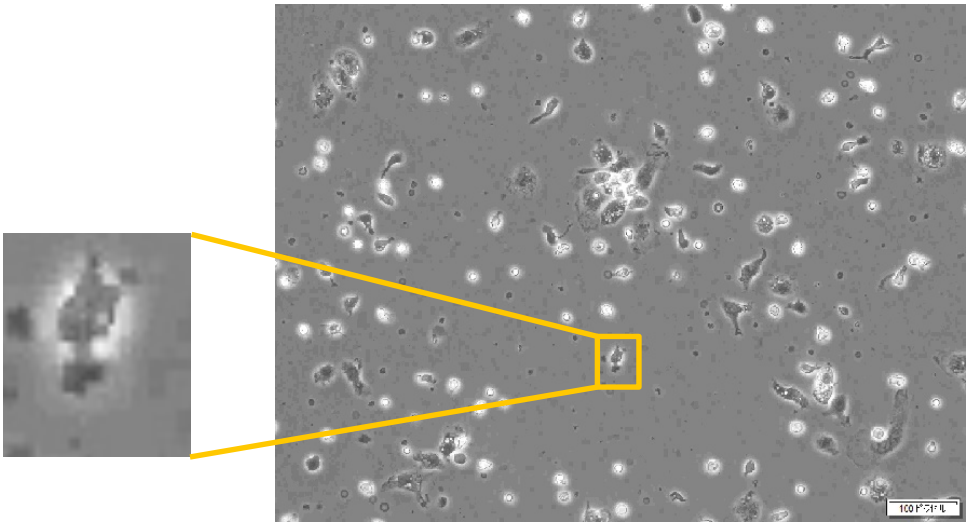


図 **1.1:** 顕微鏡画像

目的とはしていない．物体位置予測とは，画像内の状況や様子の傾向を予測するためにある程度の個数の物体を取得することを目的としている．画像内の情報を把握する上で，何の物体がどれだけ写っているかは必要な情報である．しかし，人間が画像内の様子を大まかに理解するとき，その情報は完全なものである必要はない．画像内の大体の傾向を理解できるだけで画像の内容を把握することができるからである．一般的に物体検出タスクではコンピュータによって正確に推論を行うことを目的としている．しかし，人間の意思決定の補助が目的であるとき，少なくとも人間が理解しやすいように大体の傾向を示すだけでも十分な効果があるはずである．近年の物体検出問題には CNN を用いた手法が多く提案されており，年々その精度は高くなっている．本研究においても，部分的に CNN を用いることで座標ラベルの必要ない高精度物体位置予測を目指している．

この物体位置予測を実現するために，本稿では 2 種類の物体位置予測手法を提案及び検証結果を元にした比較考察を行う．1 つ目は，予測対象の画像から固定サイズのパッチ画像をラスタースキャンで切り出し，CNN によって物体認識を行う．そして対象が認識された頻度を元に物体位置を予測するというものである．2 つ目は，予測対象の画像を学習済み Faster R-CNN [31] の内部構造である Region Proposal Network を用いて物体領域候補を取得し，それらに対して CNN による物体認識を行う．そして認識されたクラスに応じてで物体位置を予測するというものである．本研究ではこれらの手法について免疫細胞のフレーム画像を用いて実験・検証することで，座標アノテーションを必要としない物体位置予測精度の向上を目指す．

1.3. 本論文の構成

本稿では，座標アノテーションを必要としない物体位置予測の手法，及び免疫細胞のフレーム画像を用いた検証結果について述べる．本稿の構成は以下の通りである．第 2 章では本研究に関連する研究について述べる．第 3 章ではラスタースキャンを用いた物体位置予測の手法及び実験結果について述べ，第 4 章では Faster R-CNN を用いた位置予測の手法及び実験結果について述べる．そして，第 5 章では第 3 章と第 4 章の結果を元に比較，考察について述べる．最後に第 6 章では本稿についてのまとめを述べ，今後の研究課題と展望について述べる．

第2章　関連研究

2.1.　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 (CNN:Convolutional Neural Network) は近年研究が熱心に行われている深層学習手法の一つで，主に画像認識分野で優れた性能を発揮している．このアルゴリズムはニューラルネットワークが2次元の画像形状を保持したまま処理できるように拡張されたものである．LeCun らによって発表された LeNet [32] において，畳み込みニューラルネットワークの基礎形が出来上がった．CNN は主に畳み込み層とプーリング層からなる特徴抽出部分とそれらを1次元に変形し，全結合層 (ニューラルネットワーク) によって予測スコアを出力する分類部分からなる．畳み込み層は畳み込みフィルタを用いて入力画像や特徴マップから特徴抽出を行う層である．特に画像に関する畳み込み処理は二次元空間に対して畳み込みフィルタをスライドさせながら掛け合わせることで行われる．入出力のチャンネル数が1の畳み込みにおいて，畳み込みフィルタ $\boldsymbol{K}$ が画像 $\boldsymbol{I}$ に対して行う畳み込み処理を (2.1) 式に示す．

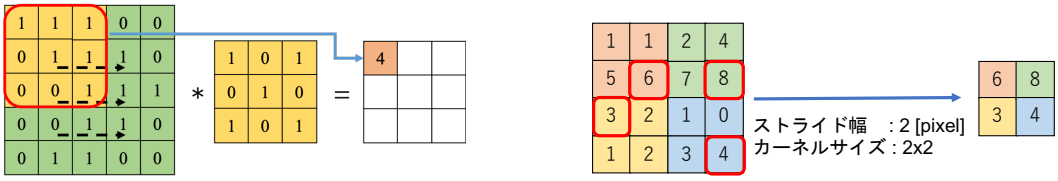
$$(\boldsymbol{I} * \boldsymbol{K})(i,j) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} I_{i+m,j+n}K_{m,n} \tag{2.1}$$

このとき，畳み込みフィルタ $\boldsymbol{K}$ のサイズを $M \times N$，入力画像サイズを $h \times w$ とし，$0 \leq i < h-N$，$0 \leq j < w-M$ とする．また，ある層の畳み込み層の入力のチャンネル数が c_{in}，出力のチャンネル数が c_{out} のとき，一般的な畳み込み層は $c_{\mathrm{in}} \times c_{\mathrm{out}}$ の畳み込みカーネルを持つ．この構造は重みを畳み込みフィルタ，入力を画像とした場合のニューラルネットワークと考えることができる．入出力のチャンネル数が c_{in}，c_{out} の畳み込みにおいて，$\boldsymbol{I}$ への $\boldsymbol{K}$ による畳み込み処理を (2.2) 式に示す．

$$(\boldsymbol{I} * \boldsymbol{K})(c,i,j) = \sum_{k=0}^{c_{\mathrm{in}}-1}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} I_{k,i+m,j+n}K_{c_{\mathrm{in}*\mathrm{c}+\mathrm{k},\mathrm{m},\mathrm{n}}} \tag{2.2}$$

このとき，$0 \leq c < c_{\mathrm{out}}$ とする．畳み込みフィルタは一般的に 3×3 や 7×7 が用いられいる．そのため出力のサイズが入力に対してフィルタサイズ -1 だけ小さくなる．畳み込みの回数を制限しないように，小さくなった分を0埋めなどでサイズ補完するパディング処理が一般的には行われている．また，プーリング処理はフィルタの範囲に対して空間圧縮を行う．特に最大値プーリングがよく用いられており，入力をフィルタサイズごとに切り出した時の最大値をその

範囲の代表として取り出し出力とする処理である．畳み込み処理を数回行って特徴抽出を行った後に最大値プーリングで顕著な特徴を保持しながら次元圧縮を行う目的でよく用いられる．畳み込み層とプーリング層についてを図 2.1 に示す．特に複数チャンネルにおける畳み込み処理の様子を図 2.2 に示す．



(a) 畳み込み処理　　(b) 最大値プーリング処理

図 **2.1:** CNN におけるフィルタ処理

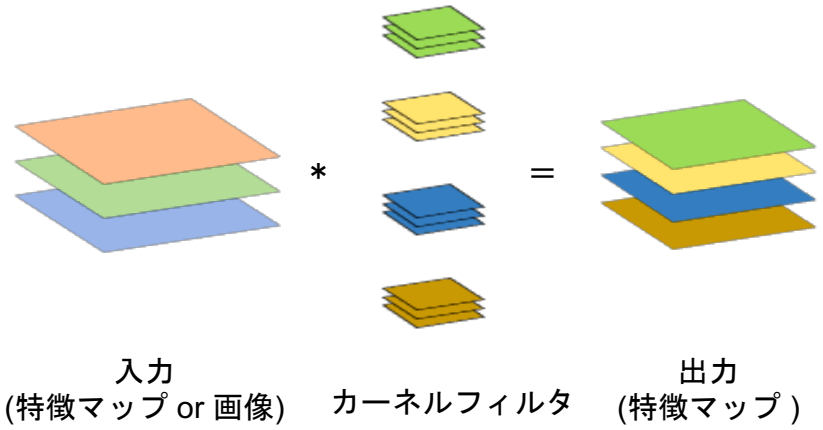


図 **2.2:** 複数チャンネルの畳み込み処理

また，プーリング層の特殊な例として Global Average Pooling 層がある．これは，　入力画像の各チャンネル平均を取り，1 次元化する処理である．つまりフィルタサイズが入力画像サイズに等しい平均値プーリングである．この処理は特に全ての畳み込み処理が行われた最後に用いられることが多く，1 次元化したのちに全結合によって出力クラスの数に調整される．GAP 層は 3 層全結合層と比べて必要なパラメータの数が少なくなり，精度の劣化もほとんどないという利点がある．この手法は最初は特徴マップの数を出力クラス数に合わせて畳み込みを行い，GAP の出力をそのまま分類スコアとする形で用いられた [33] が，近年では任意のモデルに対応できるため GAP の後に 1 回の全結合層によって出力を求めることが多い．GAP 層と全結合層を合わせた処理の様子を図 2.3 に示す．

CNN は学習パラメータを多く持つため，それらを同時に修正するために誤差逆伝播法が用いられる．誤差逆伝播法は次のステップで行われる．まず訓練データに対する予測を行い，予測結果 $\hat{y}$ を得る．次に $\hat{y}$ と正解値 y の誤差を計算する．そして，その誤差に対して 1 つ前の層

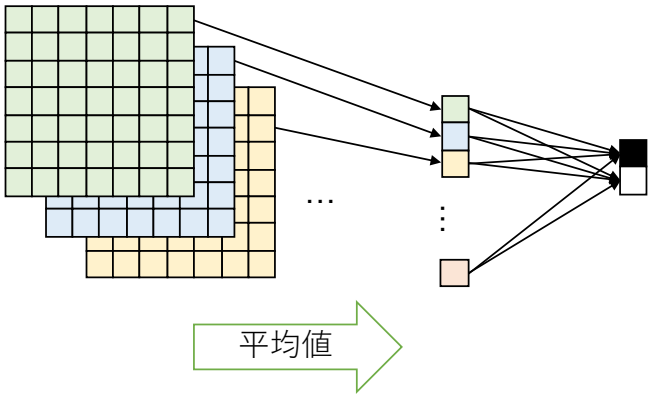


図 **2.3:** Global Average Pooling の概略図

が及ぼす影響 (局所誤差) を求める．さらに，その局所誤差に対して，その前の層からの局所誤差を求める．このように各層間の局所誤差を，ネットワークを遡りながら求めていく．その後，各局所誤差を勾配降下法によって修正する．これらを繰り返すことで重みを更新していくのが誤差逆伝播法である．また，勾配降下法とは数値的に解を見つける最適化アルゴリズムの一つである．目的関数が最小になるよう，現在のパラメータでの目的関数の勾配の逆方向へと修正することで最小値への収束を目指す．1 ステップにおける重み $\boldsymbol{w}$ の更新式を (2.3) 式に示す．この時，$E(\boldsymbol{w})$ が目的関数を表し，μ が学習率を表す．目的関数を各重みパラメータで偏微分することで目的関数の勾配を計算し，勾配と反対方向に $\boldsymbol{w}$ を更新することで目的関数の値を減少させられるとしている．

$$\boldsymbol{w} := \boldsymbol{w} - \mu \nabla_{\boldsymbol{w}} E(\boldsymbol{w}) \tag{2.3}$$

特に，多く用いられる勾配降下法としてミニバッチ勾配降下法がある．ミニバッチ勾配降下法は訓練データを無作為にミニバッチと呼ばれる訓練データの小集合に分割して行われる勾配降下法である．ミニバッチごとに (2.3) 式によって $\boldsymbol{w}$ を更新していく．ミニバッチは無作為に取得されるので，全ての訓練データによる更新に比べて不規則に変動する．そのため，目的関数が非凸関数の場合に局所最適化を抑制できる利点がある．ミニバッチのサイズをバッチサイズといい，ミニバッチの数をステップ数という．ミニバッチにおけるバッチサイズとステップ数の関係を図 2.4 に示す．

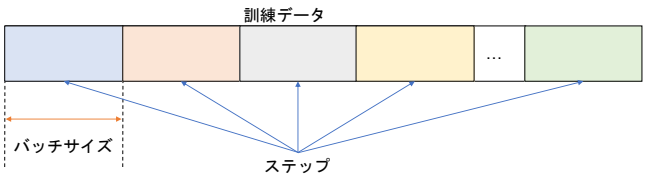


図 **2.4:** 訓練データのミニバッチ

一般に，CNN による学習は訓練データを全て使い切るまでを 1 エポックとして複数エポック

行われる．1 エポック内では訓練データをミニバッチに分割し，誤差逆伝播法による学習を行う．誤差逆伝播を行う時の誤差関数には平均二乗誤差 (Mean Squared Error: MSE) やクロスエントロピー誤差 (Cross Entrorpy Error) などのような連続関数が用いられる．これは，誤差逆伝播において勾配を用いて修正を行うのに微分可能である必要があるためである．特にマルチクラス分類では出力層に各クラス出力の合計が 1 になる SoftMax 関数をよく用いるため，誤差関数としてクロスエントロピー誤差を用いることが多い．クラス数を C，SoftMax 適用前のクラス i 出力を a_i，適用後のクラス i 出力を $\hat{y}_i$ とした時，クロスエントロピー誤差は (2.5) 式で示される．この時の $\hat{y_n}$ は SoftMax 関数の出力であり，(2.4) 式で示される．また，y_i は i 番目のクラスの正解値とする．

$$\hat{y}_i = \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp(a_j)} \tag{2.4}$$

$$E = \sum_{i=1}^{C} y_i \log \hat{y}_i \tag{2.5}$$

CNN において，高精度なモデルを作成するためにはネットワークの計算結果が広い表現力を持つことが大事であるとされており，そのためにネットワークを構成する層数を増加させることで高精度な予測ができるモデルを学習できると考えられている．そのため，VGG16 [22] では LeNet や AlexNet [19] などの従来の 5×5 の畳み込みフィルタによる畳み込み処理が 3×3 を 2 回行うことと計算量が同等であることに着目し，全ての畳み込み層で 3×3 の畳み込みフィルタを用いており，学習可能な層を当時最多の 16 層持つ．特に 2014 年の ILSVRC では第 2 位を獲得しており，現在も広く用いられるアーキテクチャとなっている．しかしながら，単純に層数を増やすと勾配消失問題や劣化問題という予測精度が低くなる現象が発生する可能性がある．これらは CNN の最適化において勾配を用いていることが原因で起こるとされている．CNN の学習時におけるパラメータの最適化には勾配降下法という手法が用いられており，予測と正解の誤差を最小点に近似させるために損失関数の勾配を用いている．この勾配は CNN の持つ各パラメータから計算されており，実計算上はネットワークの出力付近で計算される勾配からの連鎖律によって表現される．また，畳み込み計算というのは線形変換処理であるので，ネットワークの表現力をあげるために CNN の内部では非線形変換が多く用いられる．この非線形変換にはかつて (2.6) 式で示される Sigmoid 関数 (図 2.5) が用いられていた．しかし Sigmoid 関数は中心部分以外において勾配が小さく，勾配が 0 になる範囲が広い．そのため，勾配の連鎖律を計算するときに途中の勾配計算に 0 が含まれた場合に以下の層の勾配が 0 となり，勾配計算によるパラメータの修正ができなくなる．これを勾配消失問題といい，解決策として (2.7) 式に示す ReLU 関数 (図 2.6) という非線形変換を用いたりパラメータの初期値をとるランダム空間を制限したりなどがある．また，CNN モデルの各層において推論時に入力分布の偏りが入力ごとに変化する内部共変量シフトという問題があり，この問題に対して Batch Normalization を用いて学習を安定化させる手法も提案された [34].

$$y(x) = \frac{1}{1 + \exp(-x)} \tag{2.6}$$

$$y(x) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases} \tag{2.7}$$

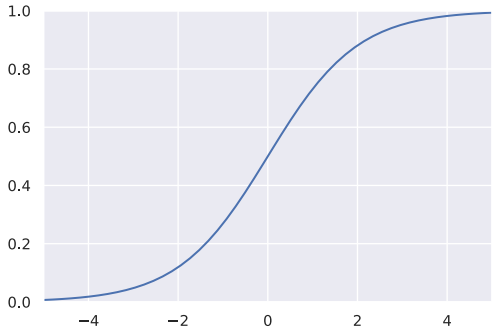


図 **2.5:** Sigmoid 関数

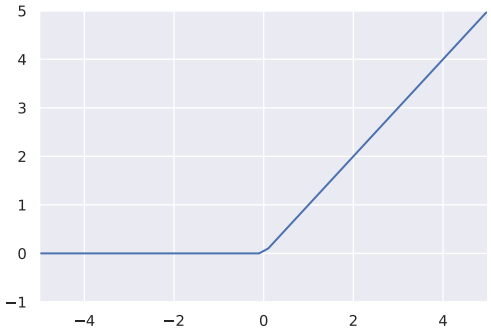


図 **2.6:** ReLU 関数

CNN の構造に大きな影響を与え，以降のネットワーク構造の基盤を作ったものの一つとして，ResNet が挙げられる．ResNet は He らによって考案されたネットワーク構造にショートカット接続をもつ CNN モデルである [23]．CNN に存在する問題として，層数増加によって学習が安定しない勾配消失問題があった．この問題は活性化関数の変更やパラメータ正規化手法によって改善したと言われている．しかしながら，He らは層を増やすと精度が劣化する問題 (劣化問題) が未だ存在するとしており，その解決策として，ある層出力の恒等写像を伝達するショートカット接続をネットワーク内部に実装した ResNet を提案した．ResNet はショートカット構造を含む Residual Block のスタックにより構成されており，block への入力は数回畳み込み層を通った後に block への入力の恒等写像にマージされる．Residual Block の略図を図 2.7 に示す．ショートカット接続では前の層の出力の恒等写像を数層先の入力にマージするため，ショートカットと並行した畳み込みを含むベースのプロセスにおいてショートカット前後の差分のみを学習することができる．ショートカットと並行したベースネットワークを $F(x, W)$ として，n 層目の入力を x_n としたとき，L 層の入力は l 層の入力を用いて (2.8) 式で表される．このとき $l < L$ とする．ここで損失関数を E とし，x_l で偏微分して勾配を計算したものを (2.9) 式に示す．ここで，$1 + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} F(x_i, W_i)$ が 0 になることはほとんどないため，勾配が 0 にならず学習が安定すると言われている．

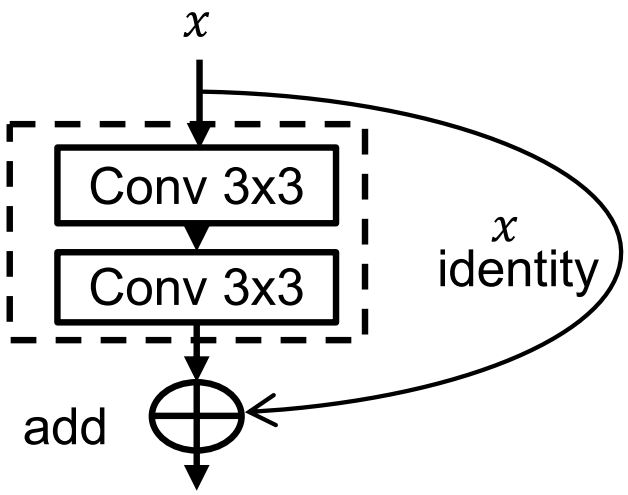


図 **2.7:** Residual Block

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i) \tag{2.8}$$

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial E}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, W_i)) \tag{2.9}$$

また，[23] で提案された ResNet は Residual Block の間に活性化関数を挟む構造になっていたが，これは必ず非線形変換を伴う活性化関数を通る形になっているため 2 つ以上の Residual Block のショートカットを超えて完全な恒等写像を伝達することができない．そこで，Residual Block においてショートカットとのマージ後に行っていた活性化をベースネットワークの畳み込み前に行うことで Residual Block の出力を直接次の Residual Block に接続した ResNet-Preact が提案された [35].

ResNet 以降，この構造をベースとしたネットワークとして MobileNet [14,15] や EfficientNet [9,36] など，さらに高精度なモデルが提案された．また，自然言語処理分野の機械翻訳タスクにおいてブレイクスルーとなった Transformer [1] の構造をベースとした Vision Transformer(ViT) [4] や，ViT の構造から Attention 構造をより単純なものへと置き換えた MLP Mixer [37] などがさらに予測精度を大幅に引き上げた．しかし，ResNet に対して近年よく用いられる最新の前処理を複数取り入れるとこで，最新のネットワークに見劣りしない精度の予測を行うことができると言われている [38]．さらに，ネットワークのベンチマークによく用いられる ImageNet [18] において，人的ミスによるラベルミスがテストデータ全体に 6%存在するとの報告がある [39]. そのため近年の高精度化したネットワークは間違ったラベルに対して過学習している恐れがあり，実際に間違ったラベルの ImageNet データで学習されたモデルを修正したデータで推論さ

せると精度が劣化したことが明らかになっている．そのなかで，ResNet は修正されたデータの推論の精度が最も高かったことが報告されている．このように，ResNet は近年のネットワークに見劣りしない精度で推論しうる拡張性を持っており，さらに頑健性も兼ね備えていることから近年再注目されている．

2.2. CNN による物体検出

物体検出とは，全体画像内においてある物体が存在する場所の座標と物体の分類を同時に行うタスクである．CNN による物体検出はさまざまなアプローチで行われており，全ての物体検出学習において，訓練画像に対して物体の座標とクラス情報がセットで付属するデータを用いて学習される．そのため，クラス予測の誤差と，正解座標に対する予測矩形 (Bounding Box: BB) の誤差を同時に最小化することでネットワークが訓練される．正解画像に対する BB の誤差は Intersection over Union(IoU) によって計算される．IoU は 2 つの領域の重なり具合を示す値であり，各領域の和集合に対する部分集合の割合である．この指標は座標情報の損失としてよく用いられる．画像内における 2 つの領域を A，B としたとき，AB 間の IoU は (2.10) 式で示される．特に A が実際の物体を示す領域，B が予測された領域とすると，IoU(A, B) は予測の正確性を示す値として解釈できる．

$$\mathrm{IoU(A,B)} = \frac{\mathrm{A} \cap \mathrm{B}}{\mathrm{A} \cup \mathrm{B}} \tag{2.10}$$

また，CNN による物体検出には two-stage モデルと one-stage モデルが存在する．two-stage モデルとは，物体検出が 2 段階によって行われる物体検出モデルである．まず 1 段階目として全体画像の中から物体らしい場所の候補を予測する．この候補を Region Proposal という．そして 2 段階目として，Region Proposal がどのクラスに属するのかを予測する．一方 one-stage モデルは物体の位置とクラスを回帰的に予測する．two-stage モデルは予測精度が高くなりやすいが予測速度が遅く，one-stage モデルは予測精度が劣化しやすい代わりに予測速度が速い．

2.2.1. two-stage モデル

物体検出に CNN を用いる例は R-CNN によって始まった．R-CNN は 2014 年に提案された CNN を用いた物体検出手法である [40]．R-CNN は Region Proposal の取得と Feature Extraction の 2 段階によって物体検出が行われる．Region Proposal の取得には Selective Search [41] を用いている．Feature Extraction は特徴抽出を意味しており，Region Proposal から特徴抽出をして，カテゴリ分類と BB のオフセットを行う．R-CNN の処理フローを図 2.8 に示す．R-CNN は Selective Search によって複数個 Region Proposal を取得し，全てを同じ大きさにリサイズする．そして物体認識用の CNN に入力し特徴抽出を行う．R-CNN では CNN による直接のカテゴリ分類及びオフセット回帰は行わず，CNN によって抽出した特徴を用いて SVM によるカテ

ゴリ分類，線型回帰による BB の座標オフセットを予測・学習する．その後，Non-Maximum Suppresion(NMS) によって BB の重複抑制を行い物体検出する．NMS とは，物体検出において重複した BB を抑制する手法である．物体検出タスクの予測結果では，同じカテゴリを予測しながら BB もほとんど被っているような重複した予測が行われることがある．NMS ではそのような予測を IoU を用いて選択的に抑制し，BB を統合する．NMS は次の手順で行われる．まず，基準となる BB を選択する．次にその BB とそれ以外のすべての BB との IoU を計算する．そして，IoU が一定よりも大きくなるような BB は取り除く．そうすることで，重複の少ない BB のみを残すことができる．また，基準の BB の決定には各 BB のもつスコアを利用することが多い．ここでスコアとは R-CNN におけるカテゴリ予測時の予測値や Faster R-CNN における Objectness などが該当する．全ての BB をスコア順に基準として見ていきつつ NMS で矩形統合を行うことで，重複した予測を制限することができる．

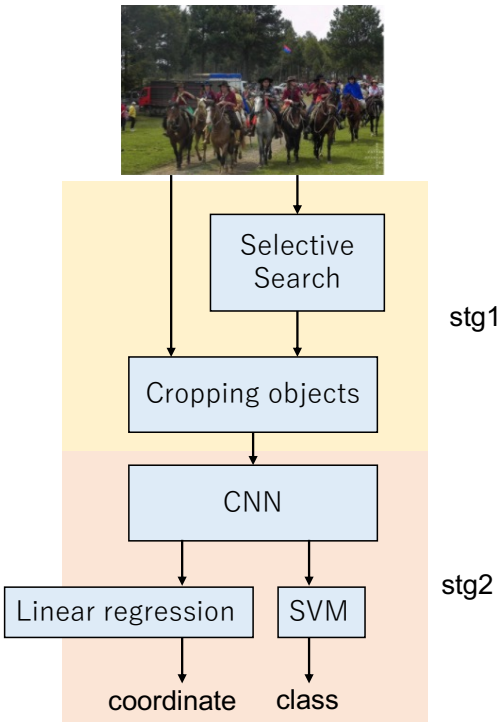


図 **2.8:** R-CNN のモデル処理フロー

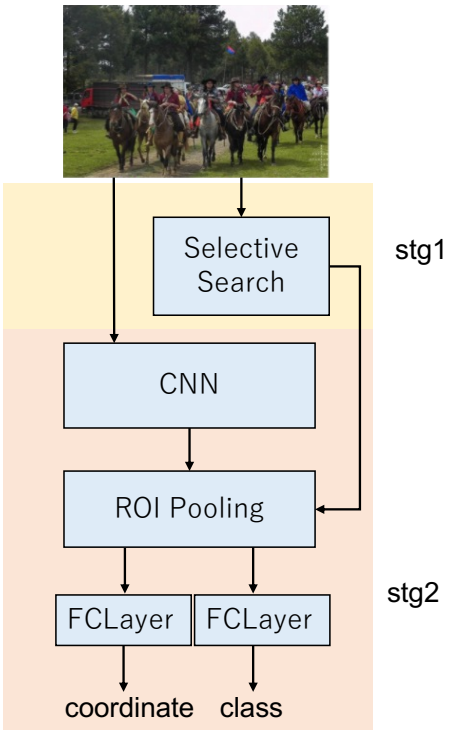
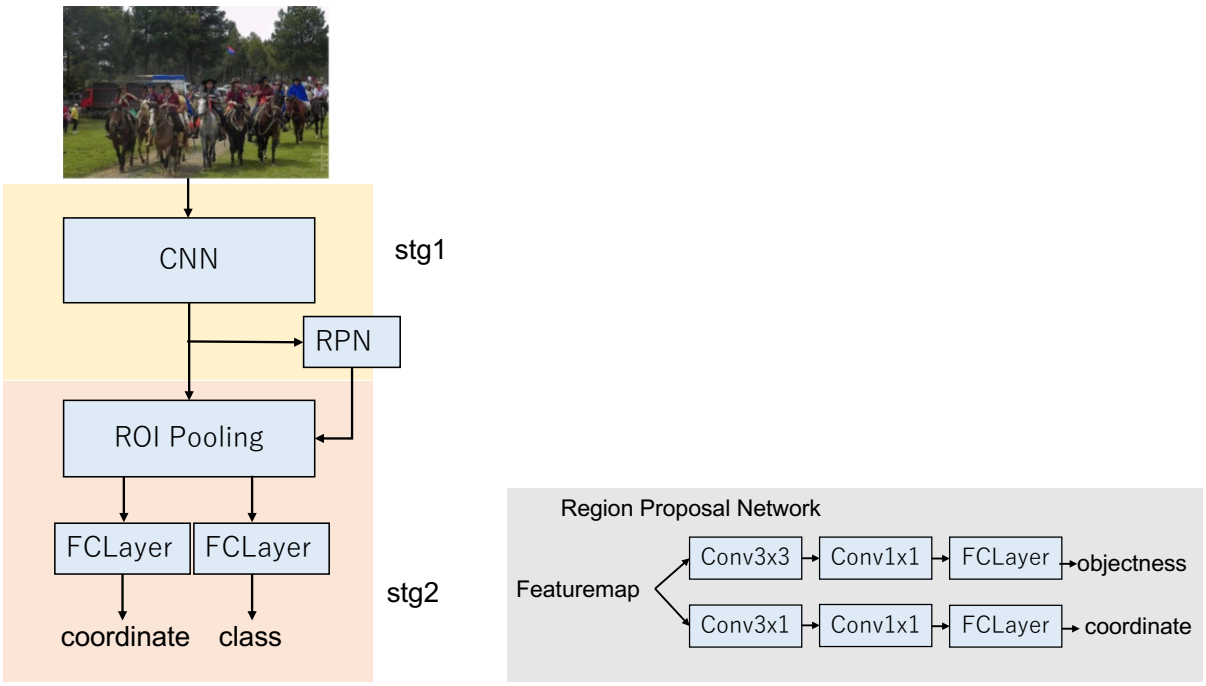


図 **2.9:** Fast R-CNN のモデル処理フロー

しかし，R-CNN は Selective Search で選択された Region Proposal は元画像から切り抜かれ，1 つ 1 つ認識用 CNN へ入力されていたため，CNN の入出力を繰り返すのは計算コストがかかり，時間がかかることが問題であった．そのため R-CNN の派生として Fast R-CNN が提案された．Fast R-CNN は予測画像全体を 1 度 CNN に入力して特徴抽出することで高速化された物体検出手法である [42]．Fast R-CNN では画像全体から一度 CNN を用いて特徴抽出を行うので時間効率が R-CNN によって高くなった．処理フローを図 2.9 に示す．Fast R-CNN における推論手順は次のとおりである．まず，R-CNN と同じように入力画像から Selective Search に

よって Region Proposal を取得する．次に入力画像は CNN によって特徴抽出された状態である特徴マップを計算する．そして，特徴マップにおける Region Proposal に該当する部分を RoI Pooling して切り出し，ニューラルネットワークによるカテゴリ予測と BB オフセットの回帰予測を行う．RoI Pooling とは特徴マップから関心領域 (Region of Interest) を取得する方法である．特徴マップは畳み込みとプーリングを介して元画像より解像度が低くなっているため，元画像とのずれを丸め込みつつ固定比率で特徴マップから取得する．以上のように Region Proposal を特徴抽出済みの形で取得できるため推論時間を短縮できている．また，予測 BB の統合のために NMS を用いている．



(a) Faster R-CNN　　**(b)** Region Proposal Network

図 **2.10:** Faster R-CNN のモデル処理フロー

その後さらに R-CNN は高速な予測が可能な形へと発展した．Faster R-CNN は Region Proposal から Feature Extraction まで 1 つのネットワーク内で行う End-to-End の物体検出手法である [31]．R-CNN や Fast R-CNN では Region Proposal の取得に Selective Search を用いており，特に Fast R-CNN では Region Proposal を CNN で特徴抽出する回数を減らすことで推論時間を短縮していたが，Region Proposal にかかる時間の短縮は行えていなかった．そこで Faster R-CNN では Region Proposal の取得を CNN で行う Region Proposal Network(RPN) を提案した．RPN は Region Proposal を予測する CNN モデルで，Faster R-CNN は RPN の前後にも CNN 構造を持っているためカテゴリ予測や BB オフセットの回帰予測と並行して学習することができる．Faster R-CNN のモデル図を図 2.10 に示す．入力画像は一度特徴抽出のために数回の畳み込みによって特徴マップが計算される．次に RPN において特徴マップに対して畳み込

みと全結合を行うことで Region Proposal を予測する．そして，特徴マップと Region Proposal を用いて RoI Pooling を行い，各 BB のカテゴリや BB オフセットを予測する．また，RPN は各 Region Proposal に対して Objectness というスコアを計算しており，Objectness の高い BB から順に基準として NMS を行なっている．

また，Faster R-CNN は Region Proposal の探索範囲の制限のために Anchor box を用いている．1 枚の画像に対する Anchor box の定義の略図を図 2.11 に示す．Anchor box はアスペクト比とサイズが固定された矩形で，RPN は複数の Anchor box の中から Region Proposal を予測する．Anchor box の定義は次のように行われる．まず元画像において等間隔に縦 H 個，横 W 個の基準点が用意される．そして各基準点において k 個の Anchor box が定義されるが，特に元論文においては 3 種類の異なるサイズで 3 種類の異なるアスペクト比の矩形を用いており $k = 9$ を採用している．そのため，定義される Anchor box の数は $H \times W \times k$ となり，これが RPN の探索空間となる．RPN は各 Anchor box に対して物体か背景かを予測し，Anchor box の座標修正を予測する．このようにして予測された Region Proposal は NMS によって統合され，物体クラスのスコア (Objectness) の高いものから N 個の Region Proposal に関して以降のネットワークでカテゴリとオフセットを予測する．

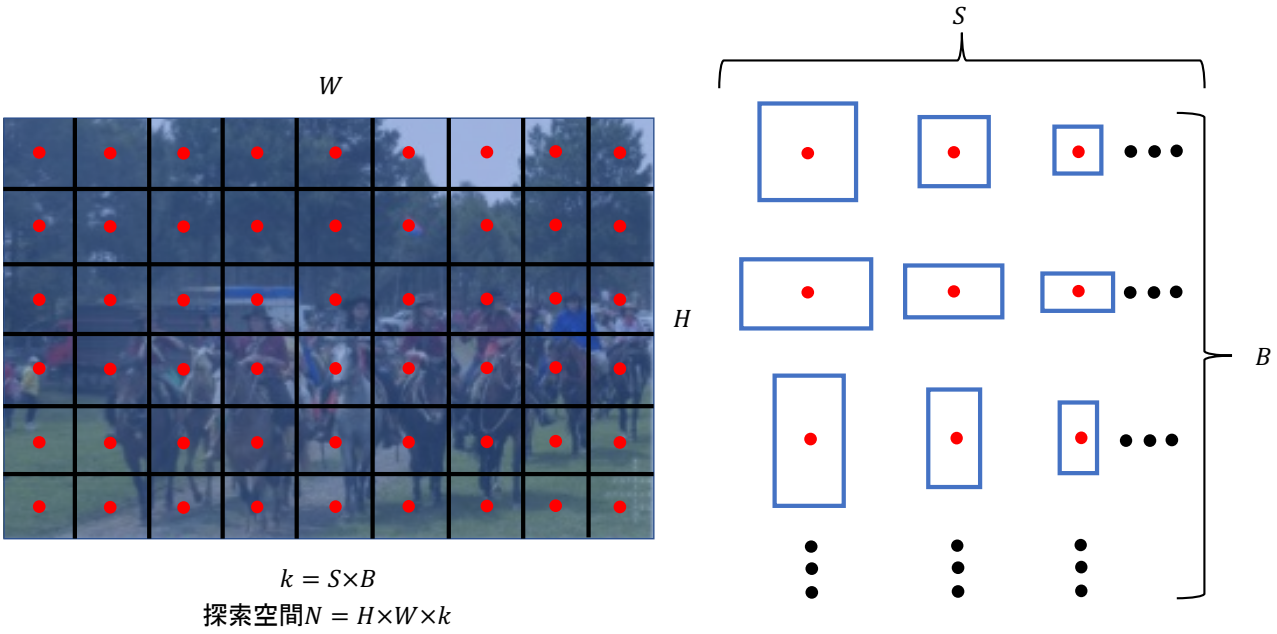


図 **2.11:** Faster R-CNN における Anchor Box

2.2.2. one-stage モデル

You Only Look Once(YOLO) [43] や Singloe Shot multibox Detector(SSD) [44] のように分類と BB の予測を同時に行う手法は one-stage モデルと呼ばれる．YOLO は Region Proposal を

行わず，カテゴリ分類と BB を直接予測する高速な物体検出アルゴリズムである [43]．YOLO は入力の画像をグリッドセルに分割し，各グリッドセル内部に中心を持つ BB を B 個予測する．各 BB の予測は，その座標情報 4 つと信頼度 $P(\mathrm{Obj})$ の計 5 つについて行われている．また，各グリッドセルが物体の一部だった場合にどのカテゴリに分類されるかの条件付きクラス確率 $P(C_i|\mathrm{Obj})$ も予測しており，(2.11) 式のように各グリッドセルに含まれる BB の信頼度との積をとることで BB のクラス確率 $P(C_i)$ を導くことができる．そして，各クラス確率を元に NMS を利用することで BB の重複を抑制し，検出予測を行う．アーキテクチャがシンプルで，探索空間となる BB が少ないため高速に処理することができる．

$$P(C_i) = P(C_i|\mathrm{Obj}) \times \mathrm{P}(\mathrm{Obj}) \tag{2.11}$$

SSD は異なる解像度ごとの特徴マップから予測を行うことで小さい物体の検出を強化した物体検出モデルである [44]．CNN は入力画像から特徴抽出を行う過程で畳み込みとプーリングを繰り返しながら特徴マップの解像度を下げていく．これは段階的に下げることで局所空間の情報を構造的に抽象化することができるためである．しかし，物体検出においてある程度解像度が下がった特徴マップを扱うことは，特徴マップの 1 画素より小さい物体の情報を他の情報と混ぜられた状態で扱うことになるため，小さい物体の検出予測を正確に行うのは難しい．SSD は CNN の段階的に小さくなる特徴マップをそれぞれ取得し，各解像度の特徴マップから物体検出予測を行うことで様々なサイズの物体を検出できるアルゴリズムである．また，BB の探索空間として各特徴マップのセルを中心とした数パターンの矩形を適用している．これは default box とよばれ，Faster R-CNN の Anchor box とよく似ている．ただし SSD は解像度の異なる特徴マップを利用するため，default box のサイズは一定でも異なるサイズのターゲットを対象にできる．default box ごとにカテゴリ分類の予測スコアと BB のオフセットを予測し，NMS で矩形重複抑制を行うことで物体検出をすることができる．

また，Feature Pyramid Network(FPN) という，階層特徴を抽出する SSD の構造によく似た特徴抽出手法も one-stage モデルとして提案されている [45]．SSD では各解像度の特徴マップからそれぞれ物体検出予測を行っていたのに対して，FPN は最終の特徴マップをアップサンプリングして一段階高い解像度の特徴マップに加算することで，大域空間まで考慮して抽出された特徴を高解像度で扱うことができる．加算された特徴マップはさらに一段階高い解像度の特徴マップに加算され，最終層で得られた特徴は前レイヤで利用することができる．FPN は Faster R-CNN の Anchor box に似た default box という矩形を定義することで探索空間に制限を設けている．Faster R-CNN において Anchor box を異なるサイズごとに数種類のアスペクト比を用意していたが，FPN を用いて異なるサイズの Region Proposal を取得することでより物体検出の精度を高めることができている．この低次元特徴を高次元特徴に埋め込みつつ解像度の高い出力を得られるピラミッド構造は Faster R-CNN にも適応できる．Faster R-CNN はサイズの違う Anchor box を最終特徴マップのみに定義していたのに対して，FPN を利用した Faster R-CNN では FPN 構造によって得られる各階層の特徴マップに対して同サイズの Anchor box を定義することで高解像度特徴を利用した Faster R-CNN として用いることができる．

2.3. CNN における予測根拠の可視化手法

CNN は学習によって獲得した重みを用いて線形変換と非線形変換を繰り返すことで特徴抽出と次元圧縮を行う．そうして得られた低次元表現された特徴をもとにニューラルネットワークによって推論を行う．しかし，推論に用いられる低次元特徴は特徴空間上で表現された特徴であり，各次元に特定の説明性を持たせることは難しい．そのため CNN は推論結果を導く過程が不透明なブラックボックスで推論の根拠を示しづらいことが問題である．この問題に対して，ネットワークのパラメータや特徴量をもとに推論根拠を可視化しようとする研究が行われている．これらは入力画像のどの部分が最終的な予測に大きく影響しているかを顕著性マップによって可視化する方法が多く行われており，さまざまな手法が存在する．

顕著性マップによる判断根拠可視化のベースとなった手法として，Class Activation Map(CAM) がある．CAM は CNN において予測の根拠となるクラス固有の特徴を顕著性マップによって可視化するアルゴリズムである [46]．CAM の処理フローを図 2.12 に示す．CAM は GoogLeNet [21] のように Global Average Pooling(GAP) [33] と 1 つの全結合層によって出力部が構成されるネットワークモデルに適応できる．クラス別の顕著性マップ L^c は (2.12) 式によって計算される．ここで A^k は最終畳み込み層における k 番目の特徴マップで，w_k^c は全結合層においてクラス c の出力の計算に用いられる k 番目の重みである．(2.12),

$$L^c = \sum_k w_k^c A^k, \tag{2.12}$$

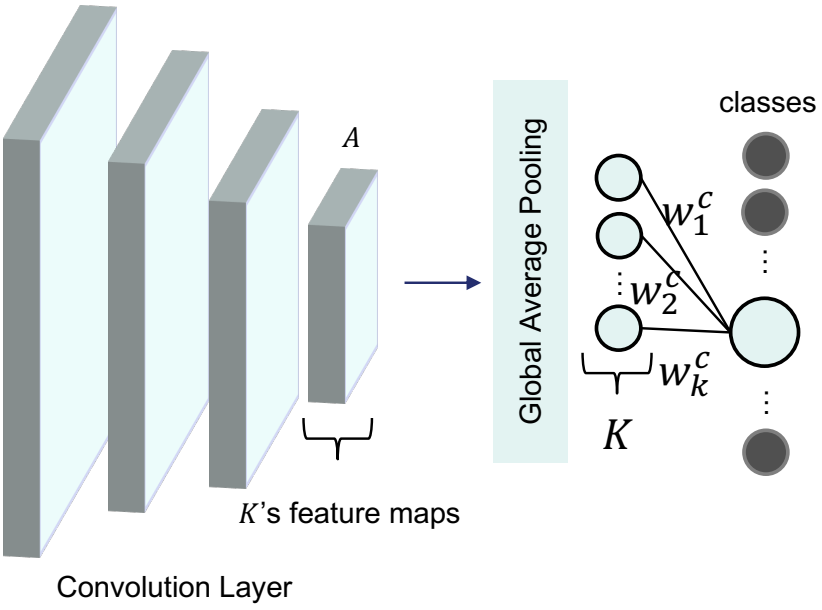


図 **2.12:** CAM の処理フロー

しかし，CAM は最終出力部分が GAP と 1 つの全結合層によって構成されているモデルにしか適用することができない．そのためそれ以外のネットワークで顕著性マップを作成するため

には出力部分を GAP と 1 つの全結合層に置き換え，数回の学習によって全結合層を学習させる必要があった．そこで，全てのネットワークに適用可能な Gradient weighted Class Activation Mapping(Grad-CAM) が提案された．Grad-CAM は CNN のクラス予測の判断根拠となる部分をネットワーク内の勾配を用いることで顕著性マップとして可視化するアルゴリズムである [47]．Grad-CAM の処理フローを図 2.13 に示す．あるパラメータが出力に影響を及ぼす場合，そのパラメータの出力に関する勾配は大きくなるため，その場所を重み付けして顕著性を可視化する．顕著性マップ L^c は (2.13) 式，(2.14) 式によって示される．このとき，A^k は最終畳み込み層における k 番目の特徴マップ，A^k_{ij} は A^k の各パラメータ，y^c は Softmax 関数を通す前のクラス c のネットワーク出力，Z は正規化パラメータとする．

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{2.13}$$

$$L^c = \mathrm{ReLU}\left(\sum_k w_k^c A^k \right), \tag{2.14}$$

各パラメータの勾配は y^c を A^k_{ij} で偏微分することで計算され，各特徴マップの勾配は各特徴マップ内のパラメータで計算した勾配の平均で表される．A^k は a_k^c で重み付けられ，k 方向に足し合わせる．そして ReLU 関数にかけることで L^c を得る．

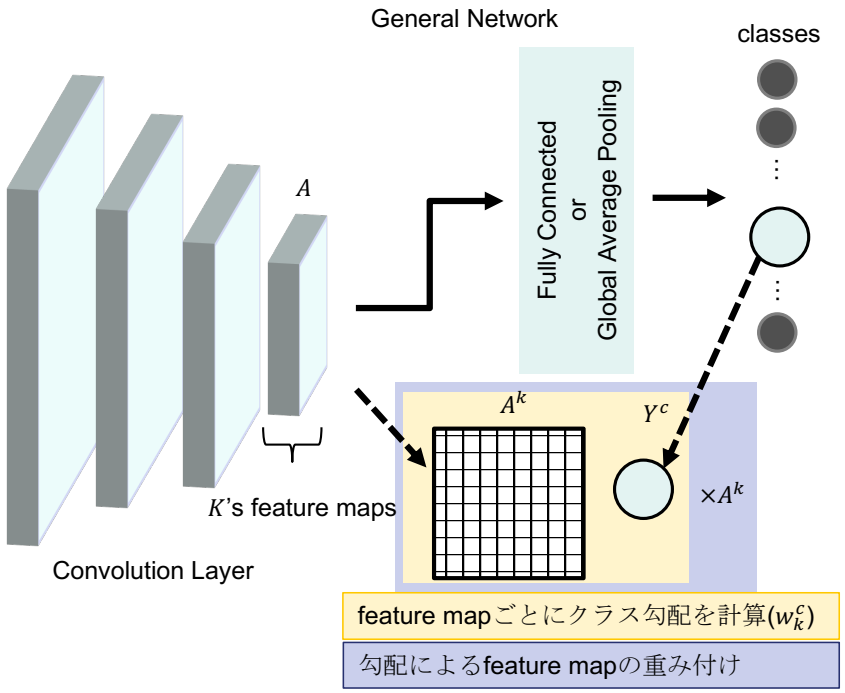


図 **2.13:** Grad-CAM の処理フロー

この Grad-CAM では，各特徴マップを特徴マップ内の平均勾配によって重み付けしていた

ため，部分的に強く反応している特徴マップを，顕著性マップにうまく反映できない可能性があった．その可能性を解消するべく提案された Grad-CAM++も，Grad-CAM と同様に勾配を用いた顕著性マップによる判断根拠の可視化アルゴリズムである [48]．Grad-CAM++の処理フローを図 2.14 に示す．Grad-CAM に対して，Grad-CAM++は微細な特徴を顕著性マップに可視化するために特徴マップの重み付けを各勾配の加重平均によって行う．顕著性マップ L^c は Grad-CAM と同じように (2.14) 式によって示される．Grad-CAM++において w_k^c は (2.15) 式によって導かれる．このとき，α_{ij}^{kc} は (2.16) 式で計算される．α_{ij}^{kc} が各パラメータで計算される勾配をもとにした荷重であり，これを用いた加重平均 w_k^c によって微細な特徴を含む顕著性マップを作成できる．

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \mathrm{ReLU}\left(\frac{\partial y^c}{\partial A_{ij}^k}\right), \tag{2.15}$$

$$\alpha_{ij}^{kc} = \frac{\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)^2}{2\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)^2 + \left(\frac{\partial y^c}{\partial A_{ij}^k}\right)^3 A_{ij}^k}, \tag{2.16}$$

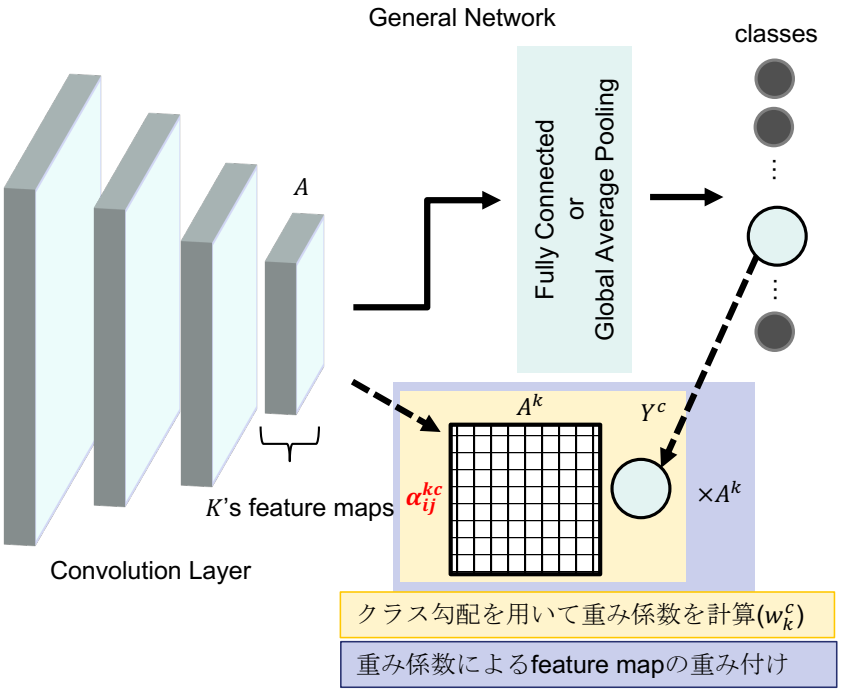


図 **2.14:** Grad-CAM++の処理フロー

CAM や Grad-CAM，Grad-CAM++のような可視化アルゴリズムにおいてはネットワーク内のクラス出力につながる重みやクラス出力の勾配を用いるため，可視化する対象のクラスを指定する必要があった．Eigen-CAM は抽出した特徴の主成分をネットワークの関心領域の顕著性

マップとして可視化するアルゴリズムである [49]．Eigen-CAM の処理フローを図 2.15 に示す．Eigen-CAM ではネットワークが抽出する特徴の上位主成分に注目し，総合的な注目領域の可視化を行う．顕著性マップ L^c は (2.18) 式で示される．ここで，V_k^1 は特徴マップの第一主成分の k 番目のパラメータで，これは (2.17) 式で示されるように A_{reshape} の特異値分解によって与えられる．A_{reshape} は各チャンネルごとに平坦化された A で，特徴マップと主成分の内積を計算することで主成分が強調されている．

$$A_{\text{reshape}} = U\Sigma V, \tag{2.17}$$

$$L^c = \sum_k V_k^1 A^k, \tag{2.18}$$

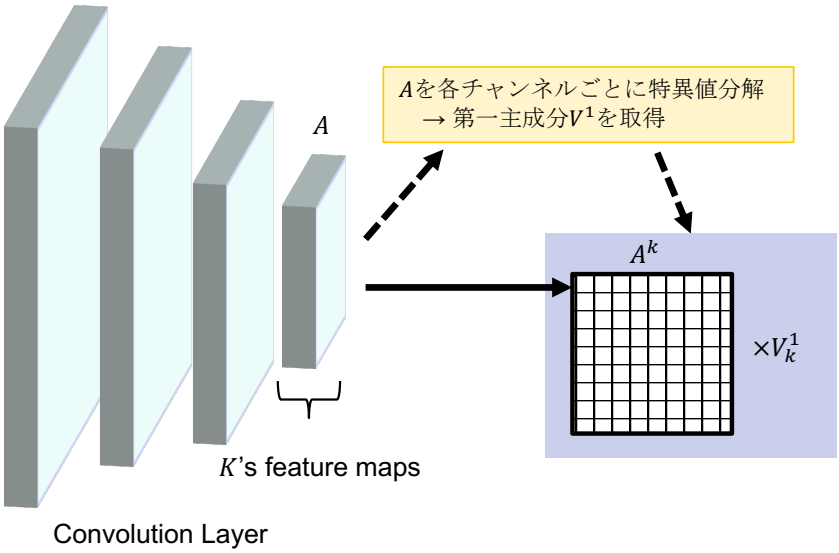


図 **2.15:** Eigen-CAM の処理フロー

第3章　ラスタースキャンを用いた位置予測

3.1.　手法

Faster R-CNN や YOLO，SSD などの CNN による一般的な物体検出手法においてはネットワークモデルの学習のための訓練データに対して「クラスラベル」と「座標ラベル」が必要となる．特に複数物体が写る画像内にクラスラベルと座標ラベルをつけるのは作業者に労力がかかり，対象によっては専門性を求められる作業である．作業を実施したとしても，ピクセルレベルの作業になるためヒューマンエラーが起こる可能性が高い．そのため，Recognition Frequency Space(RFS) を用いた座標データを必要としない学習データ，つまり物体認識用の学習データを用いて学習できる物体位置予測手法を提案してきた．

Recognition Frequency Space(RFS) は，物体位置予測を行うために，物体認識モデルを用いて複数物体が存在する画像に対して物体が存在する可能性を可視化する手法である [50]．画像内にどの物体がどの程度存在しているかという情報は，その画像がどういう状態であるかを把握する上で必要な情報である．さらにその情報は完全に把握されている必要はなく，大体の傾向を掴み，その傾向を人間が理解することで画像に対してある程度理解することができる．物体検出タスクにおいてはコンピュータの推論によって全てを正確に把握することが求められる．しかし，コンピュータの推論によって人間の意思決定を補助するという用途においてはその限りではなく，少なくとも人間が理解しやすいように大体の傾向を示すだけでも十分な効果があるはずである．そのため，本章では画像内に映る物体の数の傾向を把握するべく物体位置予測を行う．RFS は物体位置予測を行うための手法であり，CNN を用いて画像内に映る物体の存在可能性の高い位置を顕著性マップとして示すことができる．免役細胞画像に対して RFS を用いて物体位置予測を行う流れを図 3.1 に示す．

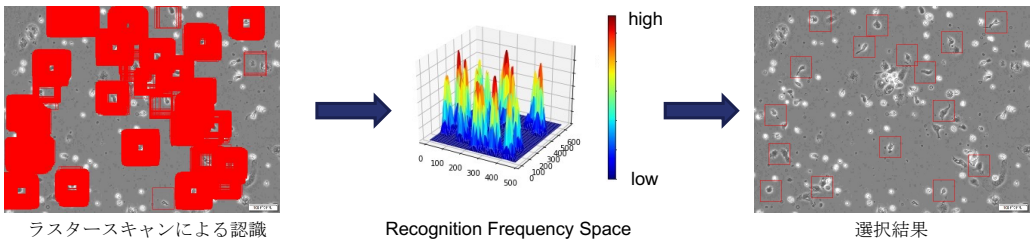


図 **3.1:** RFS を用いた物体位置予測

RFS を用いるために，事前に RFS で用いる CNN を作成する必要がある．この CNN は位置指定したいクラス c を含む多クラス分類モデルである．そして RFS は次の処理によって生成される．まず $x = w, y = h, z = 0$ の 3 次元空間 S を用意する．このとき，予測対象の画像の高さを h，幅を w とする．次に，ラスタースキャンによって固定比率のパッチ画像を切り出し，パッチ画像を CNN に入力する（図 3.2）．そしてパッチ画像が c クラスと予測された時，S 上のパッチ画像に対応する位置にガウス分布を z 方向に加算（投票）する（図 3.3）．全ての c クラスらしいパッチ画像部分にガウス分布が加算された時，S は c クラスの存在可能性が高い部分に強く重み付けされているはずである．このようにオブジェクトの存在可能性を認識頻度によって可視化した空間 S が RFS である．このときガウス分布は二次元ガウス分布を用いている．ガウス分布の中心がパッチ画像の中心と重なるように重み付けすることで画像全体に対して滑らかな重み付けを行うことができる．そして元画像における RFS の各極大点部分に対応する場所対して，固定枠の矩形を描写することで位置予測を行うことができる．

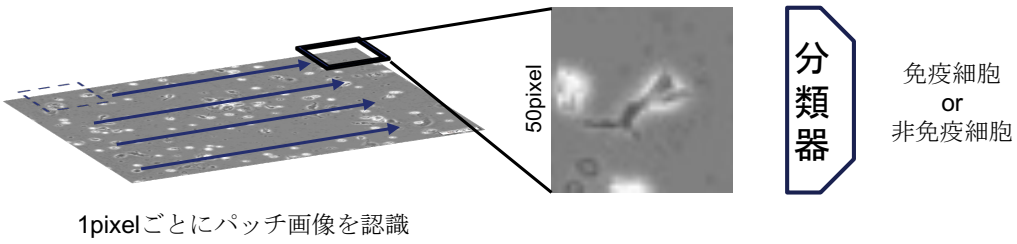


図 **3.2:** ラスタースキャンによる物体識別

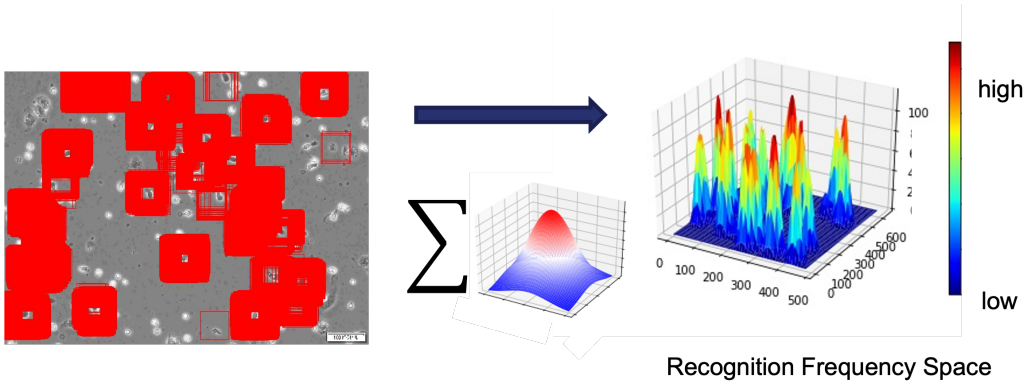


図 **3.3:** 識別結果を元にした RFS の作成

しかし，ガウス分布を用いた RFS による物体位置予測では近接する物体を別々に取得することが難しい [50]．この問題に対して，ガウス分布の一様な重み付けが原因となっていると考えた．ガウス分布は各パッチ画像の中心を原点に広がっており，パッチ画像の中心が最も強く重み付けされる．そのためパッチ画像内で物体の存在位置が偏っていた場合に適切に重み付けができない．特にラスタースキャンで切り出されたパッチ画像では，物体の場所がパッチ画像内

で徐々に遷移するため，物体の存在箇所に比べて広範囲に重み付けされる可能性がある．そこで，本章ではガウス分布による重み付けに代わり，より局所的に重み付けができる手法を用いた RFS を提案する．より局所的な重み付けとしては，図 3.4 のようにガウス分布より局所的な重み付けが可能な CNN の最終特徴マップ，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を挙げ，比較実験を行なった．RFS ではラスタースキャン時の CNN 識別において元画像から切り出されたパッチ画像を CNN に入力するため，パッチ画像に対して特徴マップや CAM などを用いた顕著性マップを計算できる．

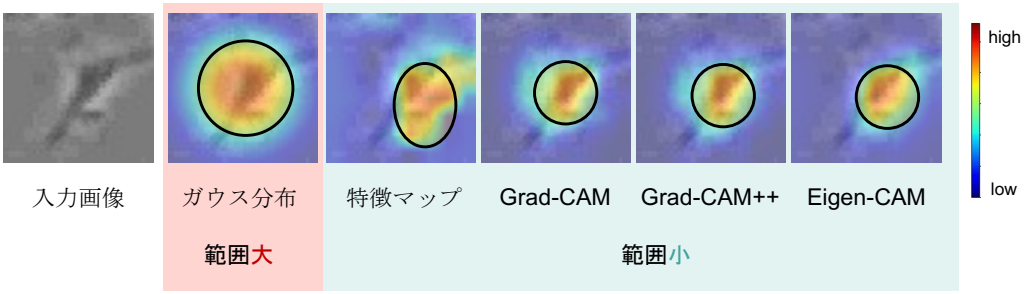


図 **3.4:** 免疫細胞画像に重み付け手法を適用した例

これらを選んだ理由は次の通りである．CAM などによる顕著性マップを用いた手法においては，物体の特徴が画像上のどこに存在するかを可視化できるため，特徴に対する顕著性を重みとして用いることで物体に対する局所的な重み付けとして用いることができると考えたためである．また，CNN の最終特徴マップは空間構造を保持したまま抽出された特徴であり，ReLU 関数を通した後の特徴は 0 以上の整数で表されることから，物体への重み付けとして有用であると考えたためである．特徴マップや顕著性マップは最終特徴マップと同じ解像度で出力されるため，これらをパッチ画像サイズにアップサンプリングすることでパッチ画像に対する RFS の重みとして用いることができる．特徴マップや顕著性マップによる重み付けは入力に対して動的に変化するため，静的なガウス分布による重み付けに比べてパッチ画像内部での対象の存在位置に柔軟に対応できる可能性がある．動的な部分重み付けによって RFS を作成する流れを図 3.5 に示す．

3.2. 実験設定

免疫細胞の位置予測を行ううえで免疫細胞の物体認識モデルを利用するため，免疫細胞と非免疫細胞を識別する CNN を作成した．CNN のアルゴリズムには PreAct-ResNet32 [35] を用いた．近年 CNN や Transformer におけるネットワークアーキテクチャの進化が盛んな中，ResNet のロバスト性や既存テクニックによる高精度性などによって ResNet が再注目されている [38,39]．そのため，本研究における免疫細胞識別は ResNet を用いて学習した．

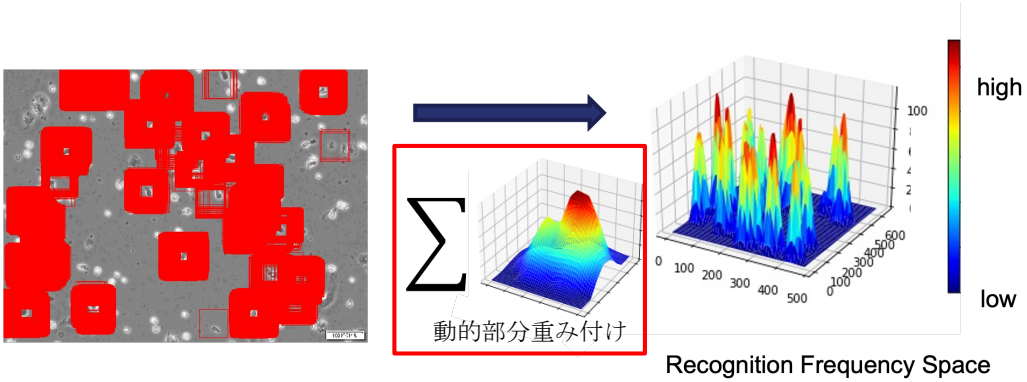


図 **3.5:** ガウス分布から動的重み付けへの置き換えが可能

この学習で用いたデータセットは 6,000 枚の免疫細胞画像と 6,000 枚の非免疫細胞画像で構成されたもので，それぞれ 5,000 枚が訓練データ，1,000 枚がテストデータとしている．データセットの画像は顕微鏡画像から切り取られたもので，顕微鏡画像の解像度は 640×480[pixel]，データセットの平均解像度は 50×50[pixel] である．全ての画像は CNN に入力される前に 32×32[pixel] にリサイズされる．この顕微鏡画像は共同研究者から提供された顕微鏡動画のフレーム画像である．そのため学習に使用した免疫細胞画像は顕微鏡動画に含まれる細胞の画像であり，これら 6,000 枚の細胞画像は変形した同じ免疫細胞を含んでいる．しかし，顕微鏡動画は 30 分ごとに撮影されたものであり，十分識別可能なほど変形しているため同じ細胞が重複しても問題ないとした．非免疫細胞画像については，免疫細胞が存在しない場所を顕微鏡動画から手作業で切り抜いた画像とした．

今回使用した CNN のモデル構造を図 3.6 に示す．このモデルを用いて 60epoch 学習を行い，Adam による最適化を行なった．学習率は最初の 5epoch の間，10^{-5} まで徐々に増加させ，20epoch 目に $\sqrt{0.1}$ 減衰，それ以降は 10epoch ごとに $\sqrt{0.1}$ 減衰させた．また，入力サイズ (高さ × 幅 × チャンネル) は $32 \times 32 \times 3$ とした．学習には 5,000 枚の免疫細胞画像と 5,000 枚の非免疫細胞画像を用いた．学習の結果，この CNN の各クラス 1,000 枚の画像を用いて性能評価をしたところ，Accuracy が 99.4%，Precision が 99.7%，Recall が 99.1%であった．Accuracy，Precision，Recall は 3.1 式，3.2 式，3.3 式で計算される．このとき，正解データを正しく正解であると予想した件数 TP（True Positive）と不正解データを誤って正解であるとしてしまった件数 FP（False Positive），正解データを誤って不正解としてしまった件数 FN（False Negative），不正解データを正しく不正解と予想した件数 TN（True Negative）とした．今回は正解データを免疫細胞画像，不正解データを非免疫細胞画像として計算した．以降の実験において，免疫細胞を識別する CNN はこのモデルを用いて実験を行なった．

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

$$Presicion = \frac{TP}{TP + FP} \tag{3.2}$$

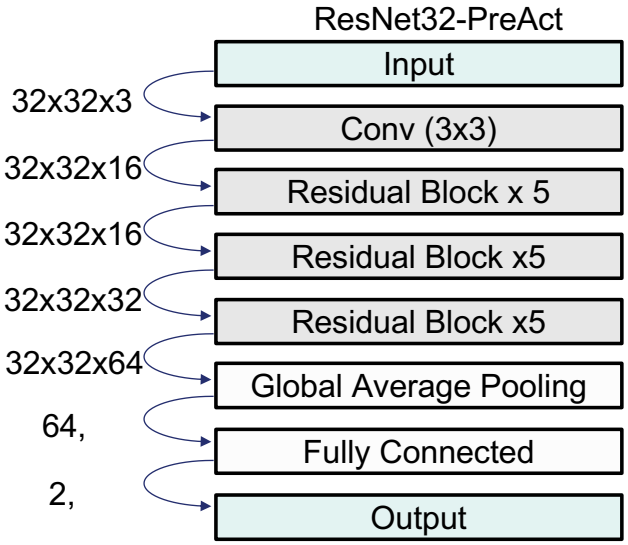


図 **3.6:** 使用した CNN モデル

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

RFS 作成に用いたガウス分布は (3.4) 式で与えられる二変量正規分布を用いた．このとき，x は任意の点を表す行ベクトル，μ は x の各要素の平均，Σ は x の分散共分散行列である．本実験では x のとりうる範囲は標準正規分布の ± 2 の切断正規分布としたため，$\mu = 0$，$\Sigma = E$ となっている．ここで標準正規分布を使用した理由は，免疫細胞はパッチ画像の中心付近に写ると仮定したからである．また RFS 内における頻度値の起伏を極端にしないために，単位行列を含む一般的な標準正規分布を用いた．

$$f(x) = \frac{1}{\sqrt{(2\pi)^2 \det\Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{3.4}$$

また，ラスタースキャンにおいて，走査の間隔は縦横 1[pixel] としており，切り抜くパッチ画像のサイズは学習画像の平均サイズと同じ 50[pixel] とした．ラスタースキャンによって得られたパッチ画像は，CNN に入力される前に 32 にリシェイプした．そして，ラスタースキャン中に免疫細胞であると認識するための CNN 出力の閾値は 95%とした．RFS からピークを取得する処理においては，ピークの探索範囲を RFS の最大値から最大値の 50%の高さの間とした．位置予測の結果として描写する固定比矩形のサイズは，ピークを中心にパッチ画像と同じ 50×50[pixel] とした．RFS 内には小さい極大点が形成される可能性があり，それは RFS を形成するときに含まれるノイズが小さくピークを形成している場所か，複数回の認識が行われていない場所である．よって，そういった場所は存在可能性も小さいと言えるため，存在可能性の高い部分のみを取得する目的で探索範囲に制限を設けた．

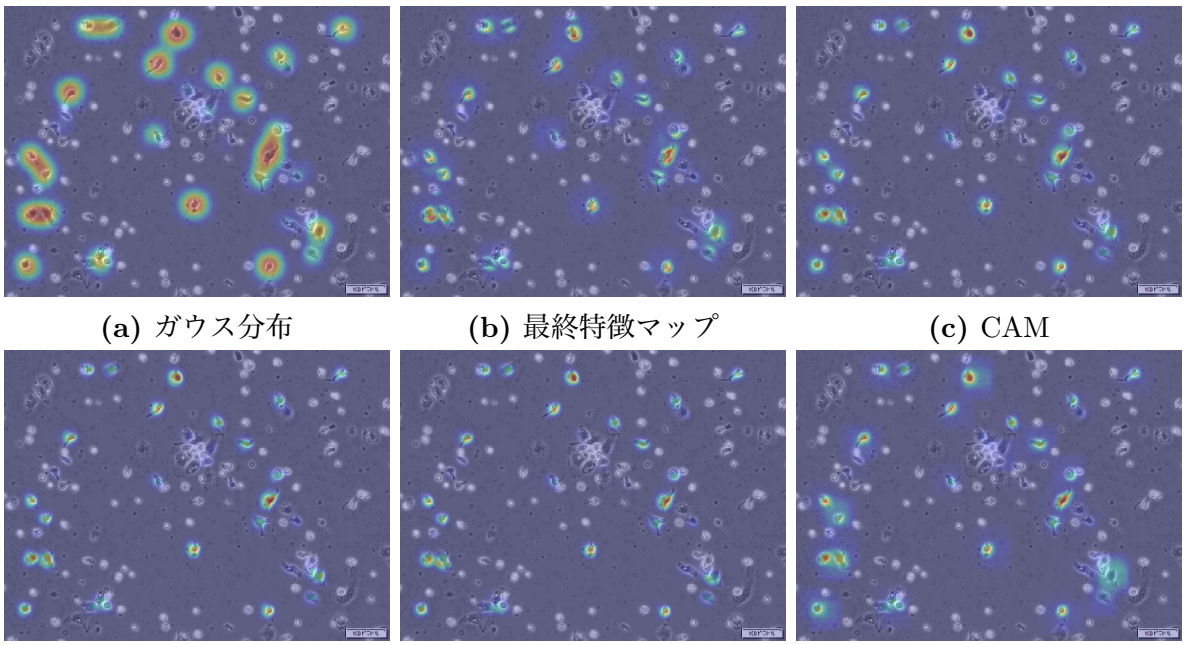
しかし，ピークの探索範囲が最大値から最大値の 50%という設定が適しているかは実験前で

は不明である．そのため，ピークの探索範囲の下限を，最大点から最小点にかけて順に小さくし，取得できたピークの個数の推移を確認した．

RFS に関するこれらの実験は，RFS の重み付け手法を変えてそれぞれ行なっており，ガウス分布，最終特徴マップ，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を用いて比較を行なった．

3.3. 結果

ガウス分布，最終特徴マップ，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を用いて RFS を作成したときのヒートマップを図 3.7，作成された RFS を用いて物体位置予測を行なった結果を図 3.8 に示す．



(a) ガウス分布 **(b)** 最終特徴マップ **(c)** CAM

(d) Grad-CAM **(e)** Grad-CAM++ **(f)** Eigen-CAM

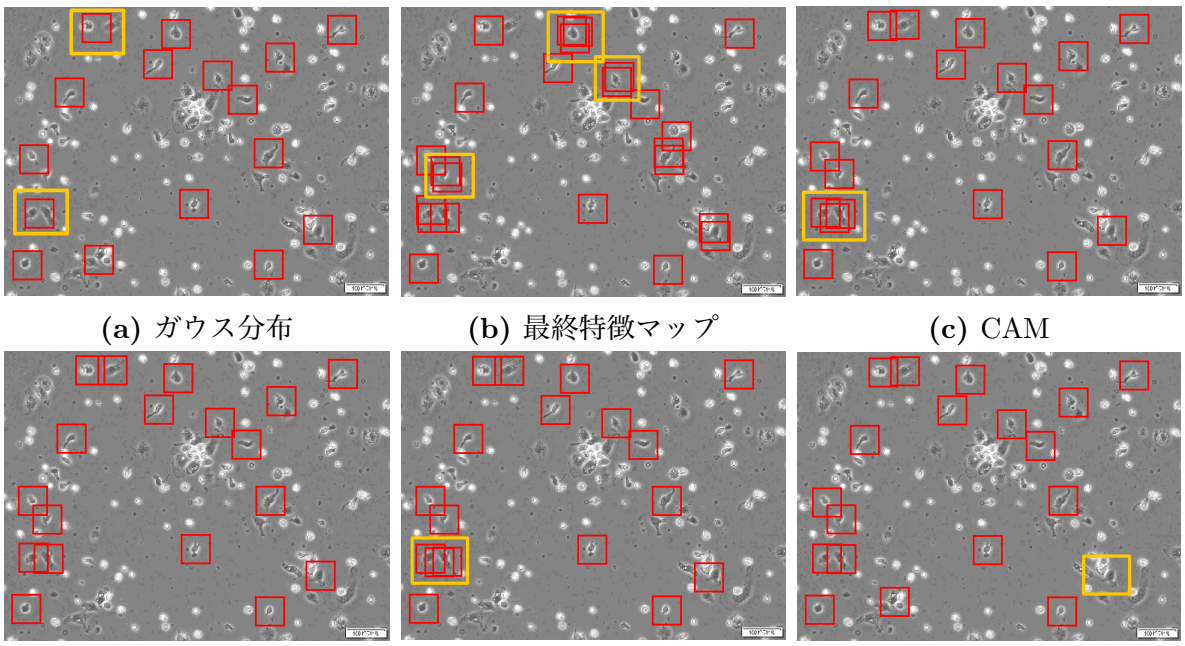
図 **3.7:** 作成された RFS

図 3.7 では強く重み付けされている部分ほど赤く，重み付けが弱いほど青く表示されている．図 3.7a は広い領域が強く重み付けされている箇所が複数あり，それらは免疫細胞の付近であった．その他の結果では，ガウス分布を用いた図 3.7a より分散の小さいピークが形成された．

図 3.8 は図 3.7 を用いて位置予測処理を行なって選択した場所に赤色の矩形を描写している．また，比較を行う上での他の手法で得た結果との代表的な違いがあった場所を黄色の矩形で囲んでいる．位置予測によって選択された個数は，図 3.7a では 16，図 3.7b では 24，図 3.7c では 19，図 3.7d〜3.7f では 18 となった．手法間の振る舞いの違いとして，図 3.7a は近接細胞を 1 つ

の矩形で囲んでいる．図 3.7b では同じ細胞を複数の矩形で囲んでいた．図 3.7c と図 3.7e では，1 つの細胞が 2 つの矩形で囲まれていた．また，図 3.7f では 1 つの細胞が選択できていなかったりなど，特に図 3.7c〜3.7f では些細な結果の違いはあったものの，選択結果として大きく変わることはなかった．

また，RFS におけるピークの探索範囲を変化させた時の選択数の推移について図 3.9 に示す．この図において，シアン色の線は図 3.8 で閾値とした 0.5 を示している．図 3.9 の左図はグラフの全体を表しており，右図は，推移の傾きを見やすくするため選択個数が 0〜60 の範囲に限定して表示している．x 軸は RFS の閾値を RFS 最大値の割合で表示しており，y 軸は選択個数を表している．最終特徴マップを用いた場合は閾値の下限が 0.6 を下回るあたりから突出して選択個数が多くなった．対照的に，ガウス分布を用いた場合は 0.6 を上回るときに選択個数が比較的多くなり，0.5 を下回るあたりから比較的選択個数が少なくなった．それ以外の手法においては推移に関して特に大きな違いはなく，0.2〜0.3，0.6〜0.8 において傾きが緩やかであった．



(a) ガウス分布　**(b)** 最終特徴マップ　**(c)** CAM

(d) Grad-CAM　**(e)** Grad-CAM++　**(f)** Eigen-CAM

図 **3.8:** 位置予測結果

3.4. 考察

まず，図 3.8 の位置予測結果について考察する．これらの結果はそれぞれの手法間の違いを示している．この結果は RFS の閾値が変化すると変わるため，図 3.9 のように閾値を変化させたときの選択個数の推移を可視化して検証を行なった．前述のように，ガウス分布を用いた RFS

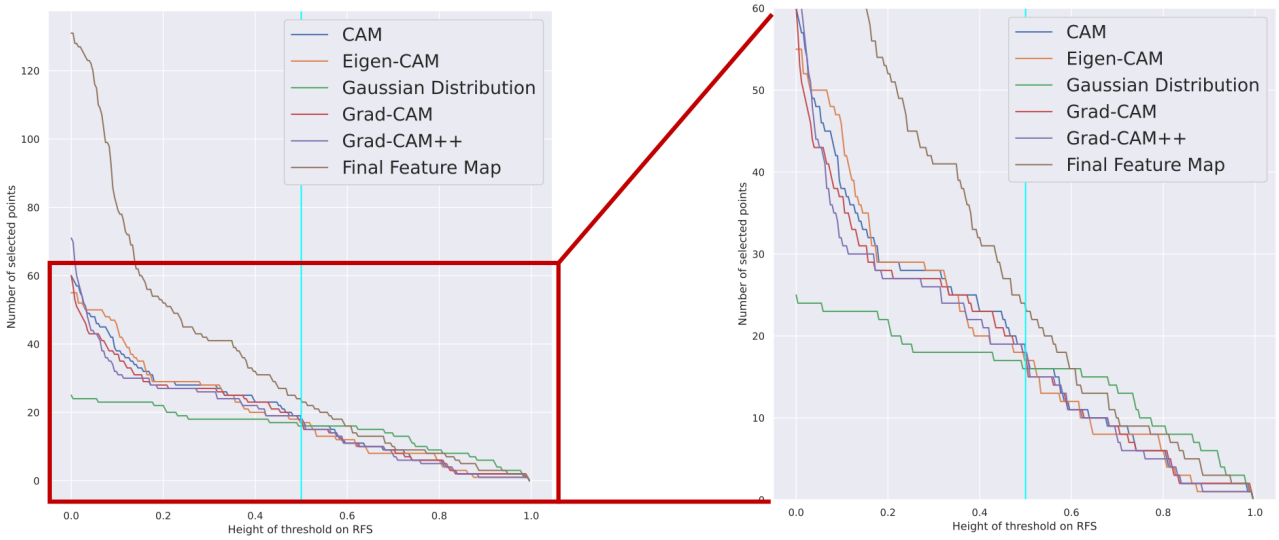
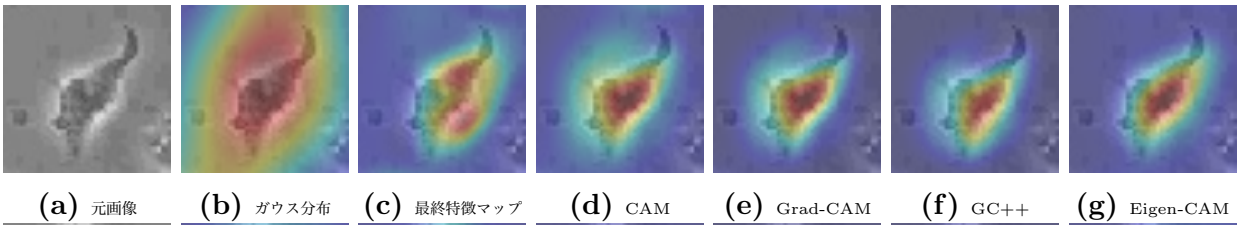


図 **3.9:** RFS の探索範囲変化による位置予測の推移

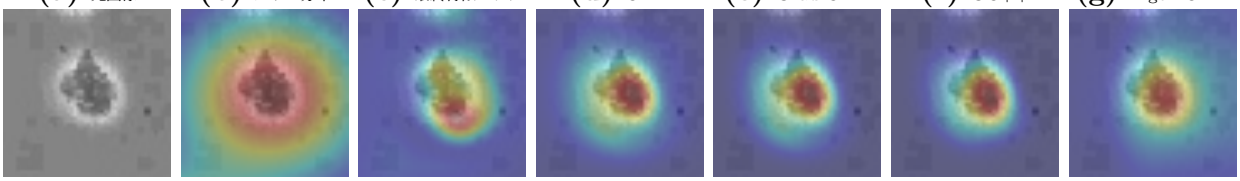
は近接する細胞を同一の細胞として取得してしまう問題があった．この現象は図 3.8a でも確認できた．図 3.9 において，ガウス分布を用いた場合の選択個数の推移のグラフは比較的滑らかであった．つまり，閾値の変化に対して選択個数は安定的であると言える．安定的であることは，その選択結果が微小な閾値の変化や他の位置予測で発生しうるノイズに対して頑健であると言える．しかしながら，この重みづけは特徴マップや CAM などのようにそれぞれのパッチ画像ごとに重みが変化せず，一定に重みづけが行われるため，近接する複数の細胞をそれぞれ分離して選択することができない．

図 3.8b に示す最終の特徴マップによる重み付けを用いた結果では，いくつかの細胞が複数の矩形で囲まれていた．これはつまり 1 つの細胞に対して複数のピークを形成する RFS ができていると考えられる．図 3.10 に RFS のヒートマップ画像を切り取ったものを示す．図 3.10c や図 3.10j のような最終特徴マップによって作成された RFS の免疫細胞への重み付けは，図 3.4 と同様な凸凹した重み付けになっている．よって，図 3.8b における選択結果の重複はこの不均一性が原因であると考える．また，この不均一性により図 3.9 に見られるような選択個数の急増が発生していると捉えることができる．よって，最終特徴マップによる選択は安定した選択を行うことは難しいと考えた．特に小さい閾値において選択個数が急激に増加しており，これは生成された RFS がノイズのような小さいピークを含んでいることが原因であると考えられる．

最終特徴マップは各チャンネルの特徴マップの平均を直接用いている一方で，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM のようなそれ以外の顕著性マップでは加重平均を用いている．この選択結果と選択個数の推移の結果はあまり違いが見られなかった．一部選択の重複などがあったものの，これらの手法は適切に対象を重み付けし，選択することができていると考えられる．単純に特徴マップを利用するだけでなく，特徴マップによって計算される重みや特徴マップの主成分は特定の特徴を際立たせるのに有用である．そしてそれらで元の特徴マップ

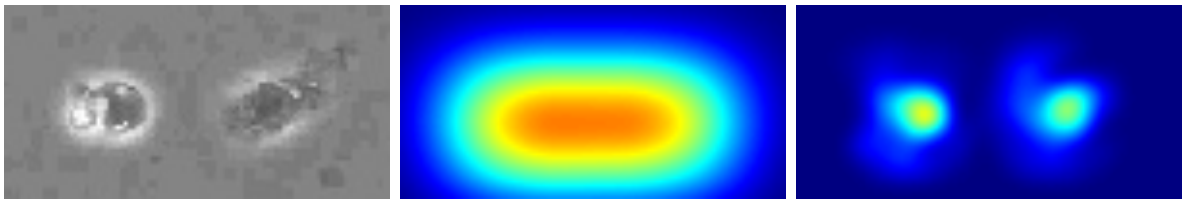


(a) 元画像 (b) ガウス分布 (c) 最終特徴マップ (d) CAM (e) Grad-CAM (f) GC++ (g) Eigen-CAM



(h) 元画像 (i) ガウス分布 (j) 最終特徴マップ (k) CAM (l) Grad-CAM (m) GC++ (n) Eigen-CAM

図 **3.10:** RFS のヒートマップ画像の切り取り画像及びその比較（Grad-CAM++のみ GC++と表示）

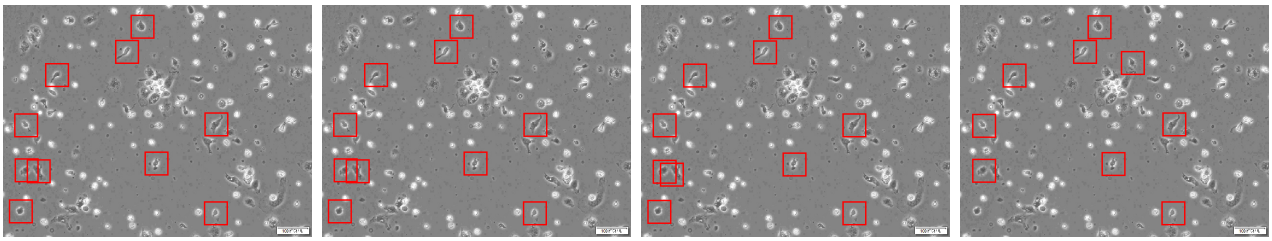


(a) 元画像 **(b)** ガウス分布 **(c)** Grad-CAM

図 **3.11:** 近接細胞に対する RFS の重み付けの違い

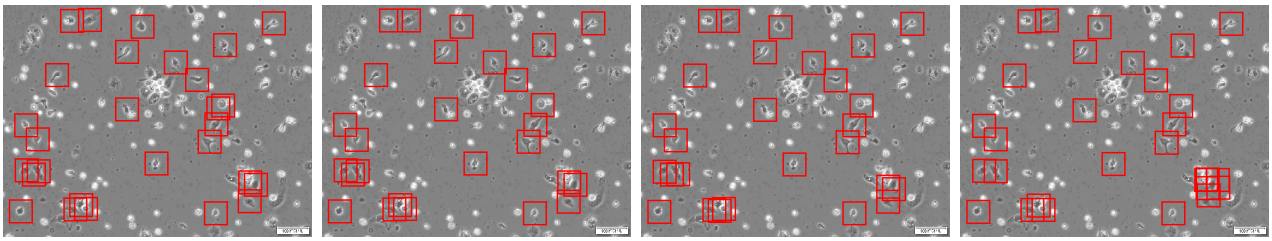
を荷重し，用いることで免疫細胞領域を強く重み付けることができる．例えば，図 3.11 に近接細胞に対するガウス分布と Grad-CAM の重み付けの違いを示す．ガウス分布を用いた例では 2 つの細胞に連なるように重み付けが行われているのに対し，Grad-CAM を用いた例では 2 つの細胞に分離して重み付けが行われていることがわかる．顕著性マップによる手法では，このようにして分離した重み付けが行えていることによって正確な選択が行えていると考えられる．また，以上の結果から，本実験で細胞選択に適している手法はなんらかの形で CNN モデルの内部出力や重みを加工して用いており，その特定の加工処理によって説明性の高い対象部分を強調することができていると考えられる．

前述のように，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM のような特徴マップの加重平均を使用する方法では，しきい値が 0.2〜0.3 および 0.6〜0.8 の場合に遷移の傾きは緩やかになる．図 3.8 の選択結果では，RFS における閾値は 0.5 と設定されており，図 3.9 の $x = 0.5$ の部分を見るとどの手法も傾きが急であることがわかる，つまり，閾値が 0.5 の時は選択個数の変化が大きく，不安定であると考えられる．図 3.12 は閾値が 0.65 の場合の選択結果を，図 3.13 は閾値が 0.25 の場合の選択結果をそれぞれ CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を用いた場合で示す．閾値 0.65 による選択数は閾値 0.5 に比べて少なくなるものの，全ての手法において全ての矩形は単一の細胞を選択できている．図 3.12a〜図 3.12c はほとんど同じ結果だったのに対し，図 3.12d では選択した細胞の場所が他手法と比べ少し異なっていた．一方，閾



(a) CAM **(b)** Grad-CAM **(c)** Grad-CAM++ **(d)** Eigen-CAM

図 **3.12:** RFS の探索範囲を 1–0.65 とした場合の選択結果



(a) CAM **(b)** Grad-CAM **(c)** Grad-CAM++ **(d)** Eigen-CAM

図 **3.13:** RFS の探索範囲を 1–0.25 とした場合の選択結果

値 0.25 を用いた場合の選択数は，閾値 0.5 の場合を超えていた．しかし，単一の細胞に対して複数の重複した選択が行われており，それは全ての手法の結果で確認された．図 3.13a〜3.13c の結果では同じ細胞を選択する複数の矩形について，non-maximum surppression などの矩形除去手法などによって取り除くことができると考えた．一方，図 3.13d の矩形は，特に左下において他の手法よりも広い範囲で重なっており，これらは non-maximum suppression で除去することは難しい．以上より，閾値を高くすると頑健な細胞選択を行うことができ，閾値を低くすると，顕微鏡画像全体における細胞を多く取得でき，全体画像の傾向を効率的に分析するのに有用である．

第4章　Faster R-CNNを用いた位置予測

4.1.　手法

ラスタースキャンを用いた位置予測における問題点として，1pixel ことに切り出した画像を CNN に入力しているため計算コストが高いことと，予測矩形が固定比で一定であることがある．ラスタースキャンによってパッチ画像を取得しているため，高さ 640px 幅 480px の画像から 50×50px のパッチ画像を取得する場合は 253, 700 枚の画像を取得することになり，それらを全て CNN によって予測する必要がある．さらに予測矩形は固定比なため，アスペクト比の変化や距離感によるサイズの変化に対応するのが難しい．特に本実験にて対象にしている免疫細胞は顕微鏡画像なため距離感によるサイズの変化はあまりないものの，免疫細胞の遊走時における形状変化によってアスペクト比が大きく変わるため柔軟な BB を予測できる必要がある．また，ラスタースキャンを用いた予測では全てのパッチ画像が Region Proposal として扱う two-stage モデルと捉えることができる．

Faster R-CNN のような two-stage モデルの物体検出予測では物体らしい領域を Region Proposal として取得しており，本実験においても同様の手法を一部利用することで柔軟なサイズの BB を用いた位置予測ができるのではないかと考えた．そこで，ラスタースキャンを行う代わりに学習済みの Faster R-CNN を用いて Region Proposal を得て位置予測を行う手法を提案する．この提案について図 4.1 に示す．Faster R-CNN は内部に背景か物体かを予測する Region Proposal Network(RPN) の構造を持っており，一般物体検出タスクによって学習済みの Faster R-CNN における RPN は背景と物体の分離に汎用的であると考えた．そのため学習済みの RPN によって Region Proposal を取得し，各 Region Proposal を CNN で識別することで，物体位置予測を行った．この手法も RFS と同様に位置情報付きのデータセットを用いておらず，物体位置を学習する必要がない．

4.2.　実験設定

本実験で用いた Faster R-CNN は一般物体検出タスク用データセットである，MSCOCO2017 [51] で学習されたものを用いた．Faster R-CNN の内部ではサイズの異なる Anchor Box を用意する代わりに Feature Pyramid Network(FPN) の構造を取り入れており，ベースラインのネットワーク構造は ResNet50 とした．この ResNet の部分と RPN の部分の学習済み重みを直接用いることで，免疫細胞の画像から objectness の高いエリアを取得し，認識を行なった．このと

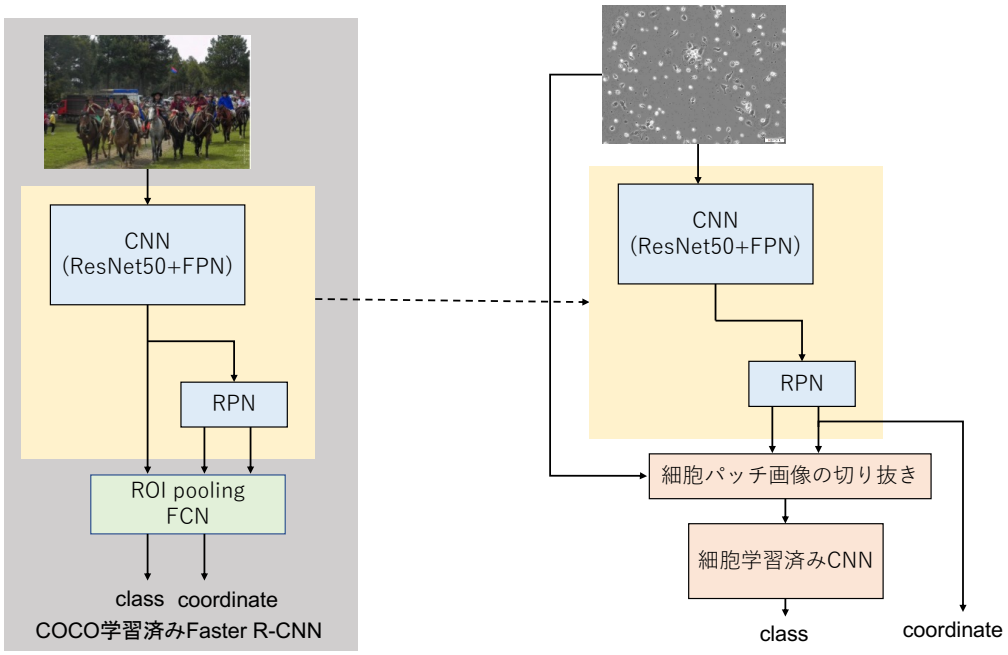


図 **4.1:** Faster R-CNN を用いた物体位置予測

き，Faster R-CNN は RPN の出力の前に NMS による Bounding Box(BB) の重複抑制を行う．細胞同士は近づきすぎると一体化してしまい分離が不可能であること，顕微鏡画像は遠近感がほとんどないため前後の重なりが起こらないことなどから，NMS の IoU 閾値は小さくしたほうが良い．そのため本実験における NMS の閾値は 0.1 と 0.3 でそれぞれ実験を行なった．

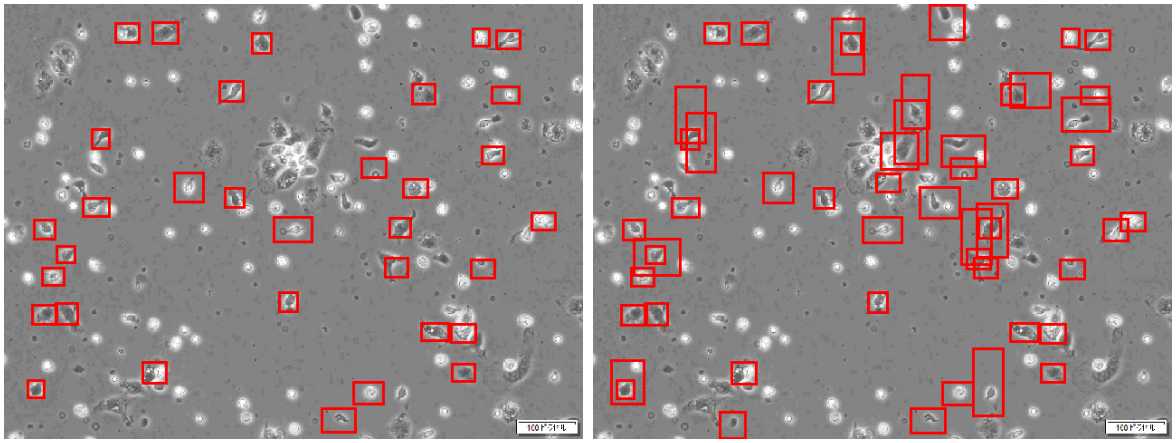
また，顕微鏡画像であることから遠近感がほとんどないため，変形は伴うものの免疫細胞はある程度の大きさが保証されている．そのため，CNN によって認識を行う探索空間を BB の大きさから制限をかけた．具体的には，CNN の学習データの平均が 50×50[pixel] であったため，一辺を $50 \pm 60\%$[pixel]，つまり 20〜80[pixel] の矩形に限定した．

RFS において，ラスタースキャンという冗長性の高いアプローチを行なっていたため CNN 出力が 95%以上で免疫細胞であると認識していたが，RPN は物体らしい場所の候補を予測するため，CNN 出力で強く制限をかける必要がない．そのため，本実験では CNN 出力が 50%以上のときに免疫細胞であると認識されるものとした．

4.3. 結果

図 4.2 に選択結果を示す．図 4.2a においては選択個数は 33 個で，IoU 閾値が小さいので全ての矩形がほとんど重複なく描写されている．また，ほとんどの細胞が正確に免疫細胞を選択できていることもわかる．一方図 4.2b においては 53 個で，図 4.2a に比べ IoU 閾値が高いので，相対的に重複して選択されている部分が多く，いくつかの場所で見られる．また，同様にこれ

らの選択のほとんどは正確に免疫細胞を選択できていることがわかる．また，これらの選択結果は全て固定比ではなく，柔軟な矩形が描写されている．横に長い形状の細胞などに，うまくフィットした矩形が用いられていることがわかる．



(a) IoU=0.1　　**(b)** IoU=0.3

図 **4.2:** RPN を用いた物体位置予測結果

4.4. 考察

用いた IoU 閾値の違いの結果を比較した場合，重複の少ない IoU=0.1 のものがうまく選択できており，免疫細胞選択には有用であるように思われる．しかし，これは IoU=0.3 と比較した結果の話であり，実験者によるパラメータの設定に大きく依存する．また，探索空間となる矩形の条件として，矩形のサイズを一辺 20〜80[pixel] としているが，このパラメータも実験結果に大きく影響する．これらのパラメータは実験者の決定依存であり，かつターゲット依存である．本実験は顕微鏡画像に映る免疫細胞がターゲットであり，顕微鏡画像は遠近感がほとんど見られない．そして免疫細胞は形状の変化はあるものの大きさはある程度の範囲で一定である．このようなターゲットに対して RPN を用いた場合，冗長な結果を多く含む可能性がある．

また，今回用いた RPN を含む Faster R-CNN は一般物体を用いて学習されたものを用いており，遠近感を考慮した推論が行われている．そのため，顕微鏡画像に映る免疫細胞のように，対象のサイズがある程度決定している場合は RPN による BB を制限するのにユーザー依存のパラメータが必要となる．

ただし，選択個数は多い点や，選択結果がおおよそ正確であることから，ユーザーによるパラメータ設定をインターフェースとして施すことで，柔軟な選択を行うことができるアプリケーションとして用いられると考えた．

第5章　考察

ラスタースキャンによってRFSを作成した場合の物体位置予測において，固定比の矩形を用いてある程度の免疫細胞を取得することができた．RFSを作成する上での重み付け手法はいくつかの手法に可換であり，比較実験の結果，CNNの中間層の出力である特徴マップの加重平均を用いる重み付けが有用であった．一方，学習済みFaster R-CNNのRPNを用いた場合の物体位置予測では，自由比なBBによって免疫細胞を取得することができた．

これらを比較したとき，RFSを用いた場合はパラメータ定義が比較的少なく，アプリケーションとして用いた場合にユーザー依存の少ない手法と言える．また，本研究でも比較実験を行なったように，中間の重み付け手法が可換であり，拡張性が高い．しかし，物体選択個数が比較的少なく，かつ固定比の矩形での選択しか行うことができない．また，選択対象のクラスが複数ある場合はRFSをクラス数分生成する必要がある．一方Faster R-CNNを用いた場合は自由比矩形で選択でき，またRPNが予測したBBを分類するだけであるため複数クラスの位置予測を同時に行うことができる．しかし，十分な予測を行うために細かくパラメータ定義を行う必要がある．さらに，本手法の出力はRPNの出力に大きく依存しており，その後の加工の自由度が低い．

第3章と第4章では，通常の免疫細胞の顕微鏡画像を使って実験を行なったが，異なるケースにおいても同様の結果が得られるかを加えて実験した．同様の条件のもと，子宮内膜症の免疫細胞画像に対して実験を行なった結果を図5.1と図5.2に示す．図5.1はガウス分布で重み付けされたRFSによって予測された結果である．また，図5.2はRPNの出力に対してIoU0.1を閾値にNMSを行なった上での予測結果である．同じ細胞を選択している矩形にはそれぞれ同じ番号をラベル付している．

選択個数としてはRFSが7番の細胞を選択しているため，RFSを用いた場合が多く選択された形となったが，それ以外は同じ細胞を安定して取得していることがわかる．また，Faster R-CNNはそれぞれの細胞に適した大きさの矩形で囲まれていることがわかる．矩形の固定比やRPN出力に対する探索空間の制限は第3章と第4章のものを引き継いでいる．そのため本研究で用いた顕微鏡画像においては，Faster R-CNNはターゲットに対して適正なパラメータを定義できれば使い回すことができることがわかる．そして，ターゲットを理解したユーザーの定義によってその性能を操作することができる．

また，本実験で用いていないデータを利用する場合において次のようなメリットデメリットが考えられる．本実験では免疫細胞のみをターゲットとして選択を行なったが，選択対象のクラスが複数であった場合，顕微鏡画像においては「赤血球」と「白血球」，一般画像において

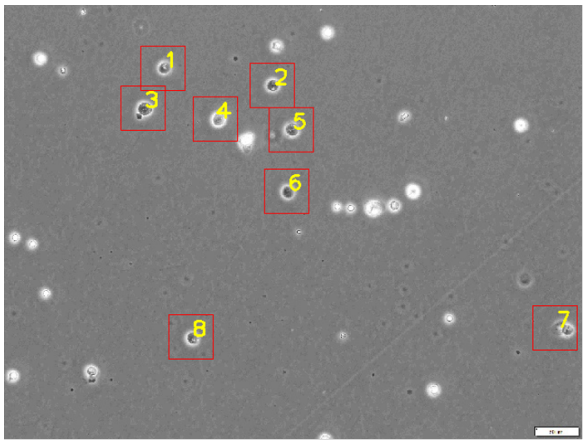


図 **5.1:** RFS を用いた実験結果

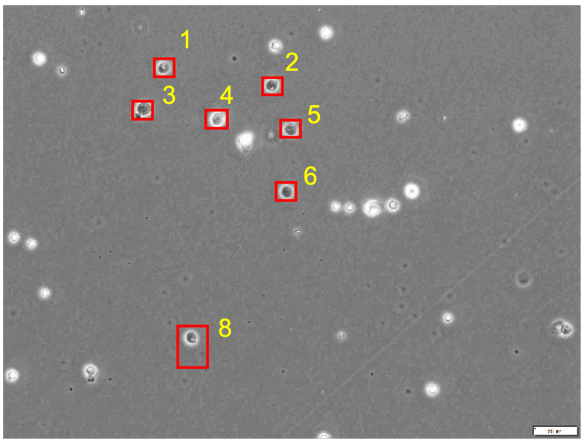


図 **5.2:** Faster R-CNN を用いた実験結果

は「人」と「猿」などを同時に選択できた方が望ましい場合がある．そのとき，RFS を用いて選択を行うと，各クラスごとに RFS を作成する必要がある．一方 Faste R-CNN を用いた場合，物体らしい場所を初めに選出し，その場所を分類するだけであるため一回の処理で同時に多クラスを選択できる．また，本研究では RFS の重み付けはガウス分布，最終特徴マップ，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を用いて実験を行なったが，この点は様々な重み付手法に可換である．重み付け手法を別手法に置き換えることで改善する可能性があるため，万能な重み付けを探索することは今後の課題と言える．対して学習済み Faster R-CNN を用いた場合は，その出力を直接用いているため依存度が高く出力の加工が難しい．また，Faster R-CNN の学習過程に依存する物体検出が行われるため，一般物体で学習された Faster R-CNN を今回のような顕微鏡画像の位置予測に用いることは難しいと考えられる．そのため，一般物体で学習された Faster R-CNN を用いるのではなく，ドメインの近い検出用データセットを用いて事前学習した Faster R-CNN を利用することで，予測精度の改善ができると考えられる．Blood Cell Detection Dataset [52] は赤血球と白血球に関する顕微鏡画像の検出用データセットであり，このようなデータセットやその他の顕微鏡画像で Faster R-CNN を学習し，学習された RPN を用いて位置予測を行うことが今後の展望として挙げられる．また，今回提案した二つの手法は互いに長所短所があり，これを埋め合わせるように組み合わせられるかについても今後考えるべきである．

また，社会における問題に対して機械学習を用いて立ち向かう場合，場面に応じたデータの収集とラベル付が必要になる．本手法が一般物体に対して応用できれば，そのような場面に応じたデータセットを作る必要がなくなり，機械学習を利用する上での一つの課題を解決できる可能性がある．そのため，今後は本手法の一般物体やその他のドメインに対しても利用できるよう，改善する必要がある．

第6章　おわりに

本研究では免疫細胞をターゲットとして，事前学習に位置情報を利用せず物体位置予測を行う手法について提案，実験を行なった．はじめに，機械学習で必要なデータセットの作成における課題について述べ，実際に免疫細胞の解析の補助となる自動システムを開発する上でも同様の課題があることを述べた．そして，位置情報を使用せずに免疫細胞を選択する手法として，ラスタースキャンを用いた位置予測と，Faster R-CNN を用いた位置予測を提案した．

ラスタースキャンを用いた位置予測とは，一枚の画像からラスタースキャンしながらパッチ画像を切り出し，それぞれを CNN で予測することで物体位置予測を行う手法である．各パッチ画像が対象であると認識された時，認識された頻度が高い場所ほど対象の存在可能性が高くなるため，頻度の高い点を物体の存在する位置として予測を行う．認識された場所に頻度として重み付けを行うことで Recognition Frequency Space(RFS) を生成することができる．ここで，重み付けには様々な手法が採用でき，本研究ではガウス分布，最終特徴マップ，CAM，Grad-CAM，Grad-CAM++，Eigen-CAM を用いて実験を行なった．その結果，特に CAM，Grad-CAM，Grad-CAM++，Eigen-CAM のような特徴マップに対して加重平均を取る手法で重み付けすることで有用な位置予測が行えた．

また，Faster R-CNN を用いた位置予測では，Faster R-CNN 内部の Region Proposal Network(RPN) という機構を利用して，物体位置予測を行った．RPN とは物体らしい位置を検出する CNN で，検出された場所を，学習済みの物体認識モデルで推論することで検出された場所にラベルを与えることができる．実験の結果，免疫細胞という対象の性質を理解した上でチューニングをした結果，高精度で多くの免疫細胞の位置を予測することができた．

ラスタースキャンを用いた場合は，固定比の矩形でしか物体位置予測を行えないが，パラメータ定義に関して頑健で，また重み付け手法において拡張性が高い．Faster R-CNN を用いた場合は，パラメータの設定がターゲット依存かつユーザー依存であり Region Proposal Network に依存度の高い予測であるが，自由比の矩形で複数クラスを同時に選択できる．特に Faster R-CNN は一般物体を学習済みのものを用いており，今回のような顕微鏡画像の実験では事前知識のドメインが異なることによる精度劣化が考えられる．

今後は，ラスタースキャンを用いた位置予測において，RFS への重み付け手法としてさらに有用なものを探索する．また，今回用いた Faster R-CNN は一般物体で学習済みのものを用いていたが，ドメインの近いデータで事前学習された Faster R-CNN を用いた場合の結果の変化についても実験，議論する必要がある．そして，機械学習を利用する際に生じるデータセットの作成という課題を解決するべく，本手法を様々なタスクに対応できるように改良を進めていく．

謝辞

本研究を遂行するにあたり，多忙にもかかわらず数々のご指導・助言を賜りました高知工科大学システム工学群電子・光システム工学教室准教授星野孝総先生に心から深く感謝致します．また，本研究のために免疫細胞解析用の体組織動画を提供してくださった高知大学医学部前田長正先生，鹿児島大学医学部牛若昂志先生に深く感謝します．そして，実験の相談を快く引き受けてくださった高知工科大学情報学群助教四宮友貴先生に心から感謝します．本システムの開発にあたり，高知工科大学システム工学群 Soft Intelligent System on a Chip 研究室の皆様には，日頃から様々な意見を頂きありがとうございます．最後になりましたが，大学・大学院生活 6 年間を支えてくれた両親・家族に深く感謝いたします．

参考文献

[1] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz; Polosukhin, Illia: "Attention is all you need", *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[2] Sutskever, Ilya; Vinyals, Oriol; Le, Quoc V: "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[3] Hochreiter, Sepp; Schmidhuber, Jürgen: "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[4] Dosovitskiy, Alexey; Beyer, Lucas; Kolesnikov, Alexander; Weissenborn, Dirk; Zhai, Xiaohua; Unterthiner, Thomas; Dehghani, Mostafa; Minderer, Matthias; Heigold, Georg; Gelly, Sylvain; Uszkoreit, Jakob; Houlsby, Neil: "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", *International Conference on Learning Representations*, 2021.

[5] He, Yanzhang; Sainath, Tara N; Prabhavalkar, Rohit; McGraw, Ian; Alvarez, Raziel; Zhao, Ding; Rybach, David; Kannan, Anjuli; Wu, Yonghui; Pang, Ruoming, et al.: "Streaming end-to-end speech recognition for mobile devices", *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385. IEEE, 2019.

[6] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina: "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Burstein, Jill; Doran, Christy; Solorio, Thamar, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

[7] Zoph, Barret; Le, Quoc V.: "Neural Architecture Search with Reinforcement Learning", *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[8] Zoph, Barret; Vasudevan, Vijay; Shlens, Jonathon; Le, Quoc V: “Learning transferable architectures for scalable image recognition”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[9] Tan, Mingxing; Le, Quoc: “Efficientnet: Rethinking model scaling for convolutional neural networks”, *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

[10] Zhao, Qijie; Sheng, Tao; Wang, Yongtao; Tang, Zhi; Chen, Ying; Cai, Ling; Ling, Haibin: “M2det: A single-shot object detector based on multi-level feature pyramid network”, *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33, pp. 9259–9266, 2019.

[11] Carion, Nicolas; Massa, Francisco; Synnaeve, Gabriel; Usunier, Nicolas; Kirillov, Alexander; Zagoruyko, Sergey: “End-to-end object detection with transformers”, *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

[12] He, Kaiming; Gkioxari, Georgia; Dollár, Piotr; Girshick, Ross: “Mask r-cnn”, *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[13] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua: “Generative adversarial nets”, *Advances in neural information processing systems*, Vol. 27, , 2014.

[14] Howard, Andrew G; Zhu, Menglong; Chen, Bo; Kalenichenko, Dmitry; Wang, Weijun; Weyand, Tobias; Andreetto, Marco; Adam, Hartwig: “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, *arXiv preprint arXiv:1704.04861*, 2017.

[15] Sandler, Mark; Howard, Andrew; Zhu, Menglong; Zhmoginov, Andrey; Chen, Liang-Chieh: “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Howard, Andrew; Sandler, Mark; Chu, Grace; Chen, Liang-Chieh; Chen, Bo; Tan, Mingxing; Wang, Weijun; Zhu, Yukun; Pang, Ruoming; Vasudevan, Vijay; Le, Quoc V.; Adam, Hartwig: “Searching for MobileNetV3”, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[17] Tan, Mingxing; Chen, Bo; Pang, Ruoming; Vasudevan, Vijay; Sandler, Mark; Howard, Andrew; Le, Quoc V: “Mnasnet: Platform-aware neural architecture search for mobile”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

[18] Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai; Fei-Fei, Li: “Imagenet: A large-scale hierarchical image database”, *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[19] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E: “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, Vol. 25, pp. 1097–1105, 2012.

[20] Zeiler, Matthew D; Fergus, Rob: “Visualizing and understanding convolutional networks”, *European conference on computer vision*, pp. 818–833. Springer, 2014.

[21] Szegedy, Christian; Liu, Wei; Jia, Yangqing; Sermanet, Pierre; Reed, Scott; Anguelov, Dragomir; Erhan, Dumitru; Vanhoucke, Vincent; Rabinovich, Andrew: “Going deeper with convolutions”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[22] Simonyan, Karen; Zisserman, Andrew: “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.

[23] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: “Deep residual learning for image recognition”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[24] Hu, Jie; Shen, Li; Sun, Gang: “Squeeze-and-Excitation Networks”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Leopold, Henry A; Orchard, Jeff; Zelek, John S; Lakshminarayanan, Vasudevan: “PixelBNN: Augmenting the PixelCNN with batch normalization and the presentation of a fast architecture for retinal vessel segmentation”, *Journal of Imaging*, Vol. 5, No. 2, p. 26, 2019.

[26] Akkus, Zeynettin; Galimzianova, Alfiia; Hoogi, Assaf; Rubin, Daniel L; Erickson, Bradley J: “Deep learning for brain MRI segmentation: state of the art and future directions”, *Journal of digital imaging*, Vol. 30, No. 4, pp. 449–459, 2017.

[27] Medus, Leandro D; Saban, Mohamed; Francés-Víllora, Jose V; Bataller-Mompeán, Manuel; Rosado-Muñoz, Alfredo: “Hyperspectral image classification using CNN: Application to industrial food packaging”, *Food Control*, Vol. 125, p. 107962, 2021.

[28] 前田長正; 泉谷知明; 楠目智章; 益本貴之; 深谷孝夫: “6-17. 子宮腺筋症の病態に関わる NK レセプターと腹腔マクロファージの免疫学的検討 (第 27 群 子宮内膜症・腺筋症 1)(一般演題)“, 日本産科婦人科學會雜誌, Vol. 56, No. 2, p. 401, 2004.

[29] 泉谷知明; 牛若昂志; 都築たまみ; 吉井智加; 谷口佳代; 前田長正: “タイムラプスを用いた子宮内膜症腹腔免疫担当細胞の動態評価 “, *Reproductive Immunology and Biology*, Vol. 32, pp. 21–26, 2017.

[30] 井上智哉; 牛若昂志; 前田長正; 星野孝総: “免疫細胞の顕微鏡画像の解析ツール開発と評価 “, 2017.

[31] Ren, Shaoqing; He, Kaiming; Girshick, Ross; Sun, Jian: “Faster R-CNN: towards real-time object detection with region proposal networks”. Vol. 39, pp. 1137–1149. IEEE, 2016.

[32] LeCun, Yann; Bottou, Léon; Bengio, Yoshua; Haffner, Patrick, et al.: “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[33] Lin, Min; Chen, Qiang; Yan, Shuicheng: “Network In Network”, Bengio, Yoshua; LeCun, Yann, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[34] Ioffe, Sergey; Szegedy, Christian: “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *International conference on machine learning*, pp. 448–456. PMLR, 2015.

[35] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: “Identity mappings in deep residual networks”, *European conference on computer vision*, pp. 630–645. Springer, 2016.

[36] Tan, Mingxing; Le, Quoc V.: “EfficientNetV2: Smaller Models and Faster Training”, Meila, Marina; Zhang, Tong, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Vol. 139 of *Proceedings of Machine Learning Research*, pp. 10096–10106. PMLR, 2021.

[37] Tolstikhin, Ilya; Houlsby, Neil; Kolesnikov, Alexander; Beyer, Lucas; Zhai, Xiaohua; Unterthiner, Thomas; Yung, Jessica; Steiner, Andreas Peter; Keysers, Daniel; Uszkoreit, Jakob; Lucic, Mario; Dosovitskiy, Alexey: “MLP-Mixer: An all-MLP Architecture for Vision”, Beygelzimer, A.; Dauphin, Y.; Liang, P.; Vaughan, J. Wortman, editors, *Advances in Neural Information Processing Systems*, 2021.

[38] Bello, Irwan; Fedus, William; Du, Xianzhi; Cubuk, Ekin Dogus; Srinivas, Aravind; Lin, Tsung-Yi; Shlens, Jonathon; Zoph, Barret: “Revisiting ResNets: Improved Training and Scaling Strategies”, *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[39] Northcutt, Curtis G; Athalye, Anish; Mueller, Jonas: “Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks”, *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[40] Girshick, Ross; Donahue, Jeff; Darrell, Trevor; Malik, Jitendra: “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[41] Uijlings, J.R.R.; Sande, K.E.A.van de ; Gevers, T.; Smeulders, A.W.M.: “Selective Search for Object Recognition”, *International Journal of Computer Vision*, 2013.

[42] Girshick, Ross: “Fast R-CNN”, *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[43] Redmon, Joseph; Divvala, Santosh; Girshick, Ross; Farhadi, Ali: “You only look once: Unified, real-time object detection”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[44] Liu, Wei; Anguelov, Dragomir; Erhan, Dumitru; Szegedy, Christian; Reed, Scott; Fu, Cheng-Yang; Berg, Alexander C: “Ssd: Single shot multibox detector”, *European conference on computer vision*, pp. 21–37. Springer, 2016.

[45] Lin, Tsung-Yi; Dollár, Piotr; Girshick, Ross; He, Kaiming; Hariharan, Bharath; Belongie, Serge: “Feature pyramid networks for object detection”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[46] Zhou, Bolei; Khosla, Aditya; Lapedriza, Agata; Oliva, Aude; Torralba, Antonio: “Learning deep features for discriminative localization”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[47] Selvaraju, Ramprasaath R; Cogswell, Michael; Das, Abhishek; Vedantam, Ramakrishna; Parikh, Devi; Batra, Dhruv: “Grad-cam: Visual explanations from deep networks via gradient-based localization”, *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[48] Chattopadhay, Aditya; Sarkar, Anirban; Howlader, Prantik; Balasubramanian, Vineeth N: “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks”, *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847. IEEE, 2018.

[49] Muhammad, Mohammed Bany; Yeasin, Mohammed: “Eigen-CAM: Class Activation Map using Principal Components”, *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

[50] 楠瀬翔也; 四宮友貴; 牛若昂志; 前田長正; 星野孝総: “免疫細胞の自動解析に向けた深層学習による解析対象の自動指定 “, 知能と情報 (日本知能情報ファジィ学会誌), 第 33 巻, pp. 560–565, 2021.

[51] “COCO - Common Objects in Context”. `https://cocodataset.org/#detection-2017`. Accessed: 2022-2-7.

[52] “Blood Cell Detection Dataset — Kaggle”. `https://www.kaggle.com/draaslan/blood-cell-detection-dataset`. Accessed: 2022-2-7.

研究業績

学会発表

1. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総．”免疫細胞の自動解析に向けた深層学習による解析対象の自動指定” 第 36 回ファジィシステムシンポジウム FSS2020 at Online. 2020.9. 口頭発表

2. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総．”深層学習を用いた免疫細胞の自動追跡手法の提案” バイオメディカル・ファジィ・システム学会第 33 回年次大会 at Online and 北九州. 2020.10. 口頭発表

3. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総．”深層学習による免疫細胞の追跡と活動量の測定” 2020 IEEE SMC Hiroshima Chapter 若手研究会 at Online. 2020.11. 口頭発表

4. Shoya KUSUNOSE, Yuki SHINOMIYA, Takashi USHIWAKA, Nagamasa MAEDA and Yukinobu HOSHINO．”Automatic Acquisition of Immune Cells Location Using Deep Learning for Automated Analysis” Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems at Online. 2020.12. 口頭発表

5. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総．”畳み込みニューラルネットワークを用いた免疫細胞の自動解析手法の実装と評価” 令和 2 年度 日本知能情報ファジィ学会中国・四国支部大会 at Online. 2021.2. 口頭発表

6. Shoya KUSUNOSE, Yuki SHINOMIYA, Takashi USHIWAKA, Nagamasa MAEDA and Yukinobu HOSHINO．”Improving Individually Selectness for Immune Cells using Grad-CAM” IEEE CYBCONF2021 at Online. 2021.6. 口頭発表

7. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総．”部分的重み付けを用いた免疫細胞選択時における近接細胞の単体選択精度の改善” 第 37 回ファジィシステムシンポジウム FSS2021 at Online. 2021.9. 口頭発表

8. Shoya KUSUNOSE, Yuki SHINOMIYA and Yukinobu HOSHINO. "Exploring Effective Channels in Fundus Images for Convolutional Neural Networks" The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics at Online. 2021.11. 口頭発表

投稿論文

9. 楠瀬翔也，四宮友貴，牛若昂志，前田長正，星野孝総. "免疫細胞の自動解析に向けた深層学習による解析対象の自動指定". 知能と情報（日本知能情報ファジィ学会誌）, Vol. 33, No. 1, pp.560-565, 2021.

10. Shoya KUSUNOSE, Yuki SHINOMIYA, Takashi USHIWAKA, Nagamasa MAEDA and Yukinobu HOSHINO. "Enhancement of the Individual Selectness using Local Spatial Weighting for Immune Cells" Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 26, No. 2, 2022 (Q4).