最適化した軽量 CNN による組織病理学的ながん検出

Optimized Light-Weight Convolutional Neural Networks for Histopathologic Cancer Detection

孫 宜蒙(京都情報大学院大学)

Yimeng Sun (The Kyoto College of Graduate Studies for Informatics)

Abstract

病理検査はがんの診断にとって決定的なものではあるが、病理専門医の目視による診断であるため、判定ミスが起こることもあり得る。正確な診断を得ることが、その後に始められる治療の成否を大きく左右しているといえる。このため、迅速かつ正確な診断方法を開発することが大きな課題となっている。近年では、人工知能による診断についての研究が盛んに行われてきた。畳み込みニューラルネットワーク(CNN)による画像診断は、その一例である。本研究は、データとなる病理写真では高解像度であるためメモリの使用量が大きくなるという問題を、分割リサイズという軽量化の手法を用いて解決した。また CNN の構造、活性化関数および最適化アルゴリズムに対して、計算実験によって有効なものを選定した。これらの実験では画像認識の正確率以外に、モデルの大きさや学習時間なども考慮した。その結果、実用的な計算量によって約93%の正確率と約98%の適合率を得ることができた。これによれば、臨床診断において、CNN モデルによる判断を先に行い、がんと判断したもののみ病理医に精査させるような実用的運用が可能になると考えられる。

The pathological examination is decisive for diagnosing cancer, and it is currently done by a visual diagnosis of a human pathologist. Thus, judgment errors may occur. Therefore, obtaining quickly an accurate diagnosis will improve the success probability for the treatment. Based on these observations, it is crucial to develop a quick and accurate diagnostic method. In recent years, research on diagnosis by artificial intelligence has been actively conducted. Image diagnosis using a Convolutional Neural Network (CNN) is such an example. As the size of each pathological photograph is too large to handle by CNN, our first step in this research is to partition and resize it into an appropriate size, which we call the light-weight model. After carefully selecting CNN structure, activation function, and optimization algorithm, numerical results show that approximately 93% of accuracy and 98% of precision have been obtained with the proposed model. Based on these results, it can be said that there is a good possibility that the method proposed in this master project will lead to practical use as an initial test for the diagnosis.

1. はじめに

1.1 日本におけるがんの現状及び対策

日本では、1981年にがん(悪性新生物)が死因別死亡率の1位となって以来、その死亡数が一貫して上昇し続け、2019年には全死亡者に占める割合が27.3%となっている(図1)[1]。すなわち、全死亡者のおよそ4人に1人はがんで死亡したことになる。また、生涯のうちにがんにかかる可能性として、男性は2人に1人、女性は3人に1人と

推測されている。2001年以降, 日本だけでなく, 世界各国においてがんで死亡する人数が増加する一 途だった。それを減らすために, がん研究をはじめ, がんの予防対策やがん医療など, 多くの努力がなさ れてきた。しかし多くの場合, がんを予防するため には個人の生活習慣の抜本的な見直しが必要である ことから, 実行するのが難しい。このため, 悪性化 する前にがんを発見し排除することが極めて重要で あるといえる。

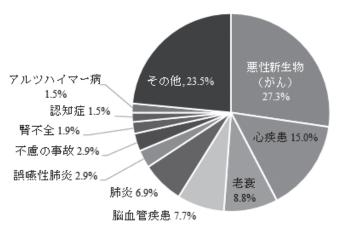
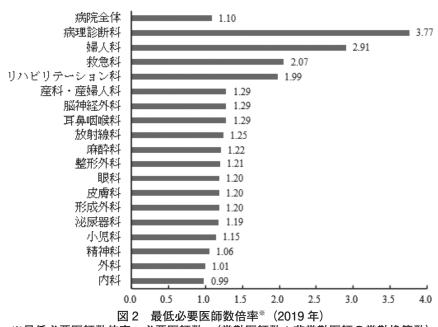


図1 主な死因の構成割合(2019年)

1.2 がんの診断

がんの診断は、検診と精密検査の2つの種類に 分けられている。がん検診とは、がんを早期に発見 し、早期に治療することを目的としている。がん検 診は一般的に、エックス線検査や検便など、非侵襲 的な検査手法が用いられる。がん検診で何か異常な 所見が検出されると、精密検査を実施することとな る。精密検査は患部によって超音波検査や MRI な どが行われるが、腫瘍が認められたら、「生検」と も言われる病理診断を行う。病理診断とは、採取し た患部の組織で病理切片を作製し、それを顕微鏡で 観察して診断することである。病理診断によって, 採取した腫瘍組織が悪性であるかどうかを判断する ことができる。近年では、血液中に存在する腫瘍マー カーなどを用いてがんを検出する研究が盛んに行わ れているにもかかわらず、がんの確定診断は病理検 杳でしかできない。



※最低必要医師数倍率=必要医師数÷(常勤医師数+非常勤医師の常勤換算数)

1.3 日本における病理専門医の現状

1.2 に記述したように、がんの病理診断は顕微鏡 で病理切片を観察する診断であり、それを行う医師 を病理専門医と呼ぶ。現在、日本では病理専門医の 数が圧倒的に不足していると言われている。たとえ ば産婦人科医と救急科医では、それぞれ最低必要医 師数倍率は 2.91 倍と 2.07 倍である。それに対し、 病理専門医の倍率は3.77倍にのぼる(図2)。この ことから、病理専門医の不足が非常に深刻なものだ と分かる[2]。病理専門医が不足していることによ り、一人の病理専門医が担当する病理診断の業務量 が格段に増加している。病理専門医の目視による病 理診断での判定ミスが起こる可能性も大幅に上昇し ていると予想されるほか、病理診断の結果が出るま での日数も増加している。正確な病理診断の結果が なければ、手術や化学療法等のがん治療を実行でき ない。つまり、正確な病理診断結果を得ることが、 その後に始められる治療の成否を大きく左右する。 以上の理由を踏まえ、病理専門医の業務を補助する ための方法を開発することが病理診断の現場で大き な課題となっている。

1.4 病理診断に深層学習を導入する研究

近年、深層学習による診断方法についての研究が盛んに行われてきた。生命科学や生物医学に関する文献を検索できるエンジンである PubMed にて、"pathology" and "deep learning" というキーワードを検索すると、2016年まで51本しかなかった報告が、2017年に107本、2018年に242本、

2019年に504本,そして2020年に650本と急増している。病理診断の分野では、深層学習の一種である畳み込みニューラルネットワーク(CNN)を用いて、採取された組織で作製された病理切片の写真を病理専門医の代わりに良性か悪性かを判断することが可能になりつつある。すなわち、まず事前に大量の病理切片を収集しCNNに学習させる。学習したCNNを用いて、新しく入力された組織写真を分析し、良性であるかどうかを判断して出力するという仕組みとなっている。

がんの病理診断に深層学習を導入 するにあたって、多くのコンペティ ションが行われてきた。そこで最も注目されたのが CAMELYON16 というものである。このコンペティションでは、乳がん患者のリンパ節の病理写真を データとし、そこにがんの転移があるかどうかを CNN に判断させる。その結果、優勝したチームのがんの認識精度は 99.4% で、11 人の病理専門医チームの認識精度の 99.6% に匹敵するものだった。

実際に、病理専門医が一つの病理診断報告を作成するにあたっては、腫瘍の悪性度以外に、組織型をはじめ様々なパラメータを考慮しなければならない。このため、病理写真を入力するだけで病理診断の報告を作らせるのは難しい。このように、深層学習による病理診断はいまだに実用化できていないが、注目株と見られる手法である。

本研究は、「作製した病理切片の写真を病理専門 医に判断させる前に、まず学習した CNN に入力し、 腫瘍があるかどうかを判断させる。腫瘍があると判 断されるものだけを病理専門医に送り、病理診断を 行う」という発想に基づいて行ったものである。

2. 病理写真の特徴

2.1 病理写真の解像度

時代とともに、病理写真の解像度は高くなっている。現在では、一般のモニターで良好な視認性を獲得するため、少なくとも 1024px × 1024px の解像度が必要である。もちろん、解像度が高ければ高いほど、視認性がより優れ、含まれる情報の量も多くなる。一方、それを処理したり送受信する際に消費するシステムリソースも多くなる。逆に、解像度の低い病理写真は、視認性が落ちるが、コンピュータでスムーズに処理できる利点がある。

インターネット上に公開された Patch-Camelyon benchmark のデータセットは、96 × 96 という非常に低い解像度の病理写真で構成されているが、CNNで学習した結果は、98%以上の認識率(Accuracy、正確に判断したもの÷データ数)が得られた。このことから、低解像度の病理写真は視認性が悪いため肉眼に不向きだが、深層学習を用いることによって正確に判別することができる。

本研究に使われる病理写真の解像度は、それぞれ 96px × 96px, 1360px × 1024px と 2048px × 1536px の 3 種類が存在する。

2.2 病理写真の特性

病理写真は一般の写真と比べて、特有の性質をいくつか持っている。本節ではそれについて簡単に説明する。

2.2.1 病理写真の色

病理写真は人体の組織の写真であるものの、コントラストを強調するため染色処理を行うため、薄紫~黒紫の色であることがほとんどである。そのヒストグラム分布の差異については図3を参照されたい。なお、染色や撮影条件の違い、また撮影機材の違いで色が変わる場合もある。

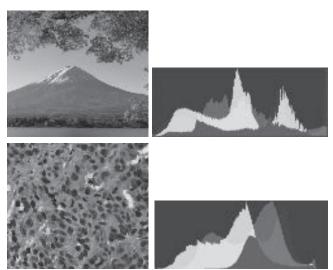


図3 一般の写真と病理写真のヒストグラム (上:一般の写真,下:病理写真)

2.2.2 病理写真の情報

病理写真は、採取した組織の病理的な状況を示すことができる(図 4)。このため、病理写真を観察する際に、一個一個の細胞の特徴だけに注目するよりも、細胞のかたまりを一つの集合体として注目すべきである。なお、細胞の形態なども組織型の判定にとって有用な情報であるため、注目する必要がある。



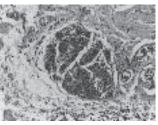


図4 本研究で用いられた病理写真 (左:腫瘍組織,右:正常組織)

3. 畳み込みニューラルネットワーク (CNN)

3.1 CNN とは

畳み込みニューラルネットワークはCNN (Convolutional Neural Network) またはConvNet とも呼ばれ,深層学習の分野で幅広く使われているモデルの一つとなっている。近年では,CNNの応用により、自然言語解析、音声認識、強化学習など様々な分野が発展を遂げた。

病理診断などで用いられる画像分類は、機械学習カテゴリの教師あり学習に分類される。入力された画像は、CNNによって分類される。CNNでは、一つ以上の層で一般的な行列乗算の代わりに畳み込みが使用される。また、CNNには、局所的な結合、共有された重み(パラメータ)、プーリング、そして多くのレイヤーの利用という4つの基本的なアイデアがある[3]。

CNN は通常、畳み込み層、プーリング層と全結合層の3段階で構成される(図5)[4]。特徴抽出段階では、まず畳み込み層が元の画像にフィルタをかけて、特徴マップを出力する。ここでは、元の数値は正規化線形ユニット(ReLU: Rectified Linear Unit)などの活性化関数を通じて計算される。次に、プーリング層でデータの圧縮を行い、意味的に似た

諸特徴を一つにマージすることができる。最後に、 集約された特徴から、全結合層を通してクラスの最 終的な判定を行う。

本研究では、認識率が高いとされている InceptionResNet 系モデル、EfficientNet 系モデル とエッジ端末に実装できる軽量の MobileNet 系モ デルを実験する。なお、これらのモデルの構成や特 徴について次の節から説明する。

3.2 InceptionResNet 系 CNN モデル

過去のCNNモデルは、3.1で紹介したように、 畳み込み層とプーリング層を順次につなぐ構造を とっていた。この構造では、CNNモデルの識別精 度を上げるのに、モデルの深さと広さを増やす手段 は有効だった。一方、モデルの深さと広さを増やせ ば、パラメータが爆発的に増大し、過剰適合などの 問題が浮上した。2014年、Google の研究者が「一 つの入力画像に対して同時に1種類のフィルタしか かけない」という観念を打破し、同時に複数種類の フィルタをかけ、その出力をマージした。このよう に、畳み込み層が一度に複数種類の特徴を学習でき る新しい畳み込み方法はInceptionと名付けられた [5]。Inceptionを利用したCNNモデルは、過去の モデルと比べて識別精度が上がり、さらに

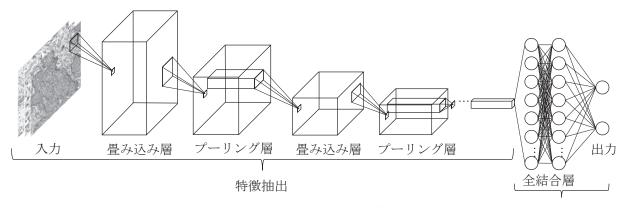


図5 CNN の基本的な階層構造

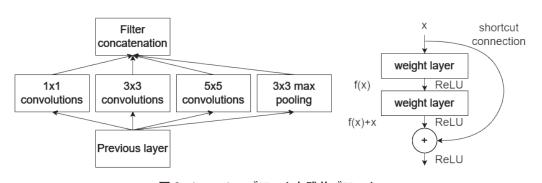


図 6 Inception ブロックと残差ブロック (左: Inception ブロック、右: 残差ブロック)

表 1 本研究に使われる CNN モデルの諸元

モデル名	パラメータ数	Top1 精度	深さ
InceptionResNetV2	55M	0.804	572
${\bf Mobile Net V1}$	4.2M	0.665	88
${\bf Mobile Net V2}$	3.5M	0.720	154
${\bf Mobile Net V3}$	5.5M	0.757	150
Efficient Net B1	5.3M	0.791	-
Efficient Net B2	7.8M	0.801	-
Efficient Net B3	9.2M	0.816	-
Efficient Net B4	19M	0.829	-
EfficientNetB5	30M	0.836	-
EfficientNetB6	43M	0.840	-
EfficientNetB7	66M	0.843	-

パラメータ増大も抑えることができた。その後、マイクロソフトが提案した Residual Network (ResNet) によって学習を加速し識別性能も上げられる残差ブロック (Residual Block) が提唱され [6]、2015年の ILSVRC の優勝モデルとなった。

InceptionResNet 系 CNN モデルは Inception と ResNet それぞれの利点を取り出し設計したモデルである。InceptionResNet 系 の モ デ ル に は、InceptionResNet と InceptionResNetV2 の二つがあるが、InceptionResNetV2 が複雑であるものの精度が非常に高かったため、一般に使われている。本研究では、InceptionResNetV2 を導入し、実験する。

3.3 MobileNet 系 CNN モデル

MobileNet 系 CNN モデルは、エッジ端末に適している 軽量 CNN モデルである。前述した InceptionResNet 系のモデルは「識別精度を上げる」ために様々な複雑な構造を導入したのに対し、MobileNet 系は、「システムリソースの節約」と「遅延時間の低減」のために作成されたモデルである。

MobileNet では Inception という処理を簡単化した bottleneck と名付け [7], それを利用する層を

depthwise separable convolution 層 と命名した。 depthwise separable convolution 層は一般の畳み込み層と比べて、同じ効果を得られながら、パラメータと計算量を大幅に削減できる利点がある。これを利用し、MobileNetV1 は通常の畳み込み処理と比べ、約 $1/8 \sim 1/9$ に総計算量を削減している。

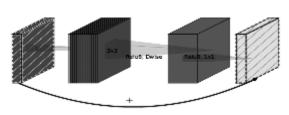
また、MobileNetV1 の発展型である MobileNetV2 は depthwise separable convolution 層に残差を導入し[8]、さらに MobileNetV3 は bottleneck ブロックに「チャネルごとの代表値を全結合層に入れ、各チャネルの重みを計算したのち最終的には元々のInput データとその重みを掛け算する」という操作を追加した[9]。その結果、計算量や遅延時間をほぼ同じに抑えながら、識別精度が上がった。

3.4 EfficientNet 系 CNN モデル

EfficientNetBX という名前の畳み込みニューラルネットワークで構成されたモデル群である。数値 X が大きいほど、ネットワーク内の重みパラメータ数が多くなるため、使用する計算資源が多くなり、画像認識の精度も上昇する。また、EfficientNet-B7 は、他のネットワークよりもかなり少ないパラメータと優れた認識精度で ImageNet 分類の最高水準を更新している。EfficientNet-B7 より下の EffficientNetも、同じレベルのパラメータを持つほかの CNN よりも精度が高いとされている。

EfficientNet のベースである EfficientNet-B0 の構造は、いくつかのモバイルボトルネック畳み込み層(Mobile Bottleneck convolution layers)から構成されており、スケーリングと一般化が容易である [10]。

ベースモデルである EfficientNet-B0 を深さと広 さ両方向にスケーリングすれば、モデルの画像認識 の精度が上がると言われている。なお、本研究の基 準モデルとして使用する EfficientNet-B6 は、ベー スモデルである EfficientNet-B0 の 1.8 倍の広さと 2.6 倍の深さを有するモデルである [10]。



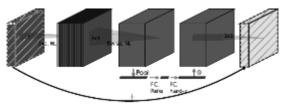
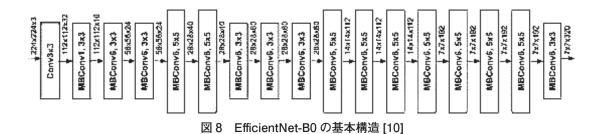


図 7 MobileNetV2 と MobileNetV3 のブロック比較 [9] (左: MobileNetV2 のブロック,右: MobileNetV3 のブロック)



本研究に使われる CNN モデルは計 10 種類であり、公開されたそれぞれのパラメータ、Top1 精度 (ImageNet データを用いて予測したラベルが実際の画像のラベルである割合) と深さは表 1 になる $^{1)}$ 。

4. 活性化関数

4.1 活性化関数とは

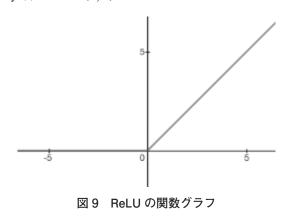
生物学的にヒントを得たニューラルネットワークでは、活性化関数を使用する。活性化関数は、上流のノードから入力を受け取り、ニューロンを発火できるかどうかを決定し、出力に変換する。また、機能の異なる層に応じて種類の違う活性化関数を用いる必要がある。シグモイド(Sigmoid)やソフトマックス(Softmax)などの関数は通常、クラスを予測することを目的とする最終層で用いられる。それに対して、ReLU(Rectified Linear Unit)や Swishなどの関数は、ニューロンの入力を出力に変換し、中間層に用いられる。

4.2 正規化線形ユニット

(ReLU: Rectified Linear Unit)

ReLU 関数は下記の通り定義される:

$$f(x) = \max\{0, x\}$$



1) https://keras.io/ja/applications/

ReLU [11] は現在、世界で最も使われている活性 化関数である。それは、区分線形関数であるため、 計算するのが容易なためである。また、入力が 0 以 下の場合は、出力はすべて 0 になるのも、ReLU の 特徴である。図 9 に示すように、座標点 (0,0) を 基点としてランプ型曲線のグラフになるため、「ラ ンプ関数」とも呼ばれている。

ReLUを使う主な利点として、除算や指数計算が一切なく、出力は0もしくは入力値のままのため、高速で計算できる。その反面、ReLUは、学習中に一部の勾配が消滅する可能性があるという欠点も持っている。勾配が消滅することにより、一部のニューロンが機能しなくなり、入力した値に関わらず0を出力するようになる。機能しなくなったニューロンが学習の妨げとなり、学習効率が落ち精度が下がる可能性もあり得る[12]。なお、この欠点を解消するため、ReLUを少し変形してPReLUやLeaky ReLU などが開発された。

4.3 Swish

Swish 関数は下記の通り定義される:

$$f(x) = x \cdot sigmoid(\beta x)$$
$$sigmoid(\beta x) = \frac{1}{1 + e^{-\beta x}}$$

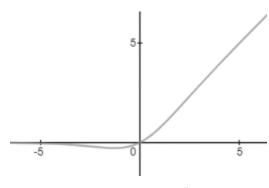


図 10 Swish の関数グラフ

Swish [13] は ReLU とは異なり、滑らかで非単調な関数である。入力値が0以下の場合は出力値がほぼ0で、入力値が0以上の場合は入力値とほ

ぼ同じ値となる性質は、ReLUと似ている。Swish は ReLU の代替として考案され、ニューラルネットワークの隠れ層の活性化関数として、現在幅広く使われている。また、複数のデータセットを用いた実験により、Swish は ReLU よりも優れた活性化関数であることが示されている。本研究に使われる Efficient 系 CNN モデルでは、Swish が ReLU に代わりデフォルトの活性化関数になっている。

なお、Swish の定義の中の β は変数である。 β = 0 の時は $f(x) = \frac{1}{2}x$ となり、 $\beta = \infty$ の時は f(x) は ReLU となる。本研究に用いられる活性化関数 Swish では、 $\beta = 1$ とする。

4.4 Mish

Mish 関数は下記の通り定義される:

$$f(x) = x \cdot \tanh(t(x))$$

$$t(x) = \ln(1 + e^{x})$$

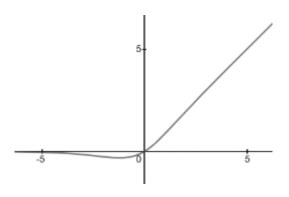


図 11 Mish の関数グラフ

Mish [14] は 2019 年に新たに提唱された活性化関数である。入力値が 0 以下の場合は出力値がほぼ 0 で、入力値が 0 以上の場合は入力値とほぼ同じ値となり Swish と似たグラフを有しているが、定義式は Swish よりも複雑である。Mish も Swish と同じく ReLU の代替として考案されたニューラルネットワークの隠れ層の活性化関数である。提唱者の論文によれば、Mish は Swish よりも精度が高いとされている。

5. 最適化アルゴリズム

5.1 最適化アルゴリズムとは

ニューラルネットワークをはじめ、すべての機械 学習は損失(Loss)を最小にするのが目標である。 そこで、効率よく損失を最小にするアルゴリズムを 「最適化アルゴリズム」と呼ぶ。ほとんどの深層学 習アルゴリズムには最適化アルゴリズムが含まれている。その中の勾配降下法 [15] は、最適化のための最も知られたアルゴリズムの一つである。勾配降下法は、モデルのパラメータによってパラメータ化される目的関数を最小化するための方法であり、目的関数の勾配方向の逆方向にパラメータを更新することによって実現される。また、Epoch(すべてのデータを一回だけ学習する)ごとにそのパラメータが更新され、最適化アルゴリズムのバリデーションによって更新量が異なる。

5.2 最適化アルゴリズムのバリデーション

確率的勾配降下法(SGD, Stochastic Gradient Descent)[16] は各学習エポックに対してパラメータを更新する。具体的には、ランダムに選択されたデータのサブセットから計算された推定値で実際の勾配を置き換える [17]。また、SGD では、すべての例のコスト関数の勾配を加算する代わりに、一つの例のコスト関数の勾配を計算する。本研究では、SGD の学習率を 0.01、モーメンタムを 0 とする。

AdaGrad [18] は学習率をパラメータに個別に適応させる。更新頻度の低いパラメータに対して更新量が大きくなるよう、頻繁に更新するパラメータに対して更新量が小さくなるように調整する [19]。AdaGrad は一部の深層学習モデルで良好に機能することが分かっている。本研究では、AdaGrad の学習率を 0.01 とする。

Adadelta [20] は一次情報のみを使用し、時間とともに動的に適応し、確率的勾配降下法の計算コストを最小限に抑える。Adagrad と比較すると、学習率が撤廃されたため、使い勝手が格段に上がった。なお、その性能も AdaGrad や SGD に劣らないようで、パフォーマンスの良い選択肢になり得る。本研究では、Adadelta の学習率を 1.0、 ρ を 0.95 とする。

RMSProp [21] は学習率を二乗勾配の指数関数的に減衰する平均で除算することから、凸関数に適用されると、急速に収束するように設計されていることが分かる。また、動的平均を使用し、その長さのスケールを制御する新しいハイパーパラメータが導入されている。本研究では、RMSProp の学習率を0.001, ρ を 0.9 とする。

Adaptive Moment Estimation (Adam)[22] は RMSprop の改良版とも言われ、各パラメータの適応学習率を計算する。Adam は、現在最も使用され

表 2 異なる最適化アルゴリズムの更新量の比較

アルゴリズム	更新量
SGD	$g_t = \nabla_{\theta_t} J(\theta_t)$ $\nabla \theta_t = -\eta \cdot g_t$ $\theta_t = \theta_t + \Delta \theta_t$
Momentum	$\Delta\theta_t = -\gamma v_{t-1} - \eta g_t$
NAG	$\Delta\theta_t = -\gamma v_{t-1} - \eta \nabla_{\theta} J(\theta - \gamma v_{t-1})$
Adagrad	$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \eta}} \odot g_t$
Adadelta	$\Delta\theta_t = -\frac{RMS[\triangle \theta_t]_{t-1}}{RMS[g]_t}g_t$
RMSProp	$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \eta}}g_t$
Adam	$\Delta\theta_t = -\frac{\eta}{\sqrt{\widehat{v}_t} + \epsilon} \widehat{m}_t$

ている最適化アルゴリズムである。また. Adam に は運動量は勾配の推定値として直接組み込まれてい る。また、原点での初期化を考慮するため、勾配と 二乗勾配の不偏推定量に対するバイアス補正も含ま れている。これは、AdaGrad の利点を利用してス パースグラジエント (sparse gradients) を処理し, RMSProp に非定常目標を処理する機能を組み合わ せている。学習率は時々提案されたデフォルト値か ら変更する必要があるものの、一般的にハイパーパ ラメータの選択に対してロバストであると考えられ ている。本研究では、Adam の学習率を 0.001、β₁ を 0.9. β₂ は 0.999 とする。

Adamax [23] は、Adam を無限次元ノルムに対 応させたものである。過去の勾配の指数移動平均よ りも大きい勾配が入力される場合, Adamax が保 持した平均勾配をクリアし、最新の勾配絶対値に更 新するのが、Adamax の特徴である。本研究では、 Adamax の学習率を 0.002. B₁ を 0.9. B₂ は 0.999 とする。

Nesterov-accelerated Adaptive Moment Estimation (NAdam)[24] は、ネステロフの加速法

を Adam に取り入れたものである。NAdam は、ノ イズの多いグラデーションまたは曲率の高いグラ デーションによく用いられる。また、NAdam の学 習プロセスは、前の勾配と現在の勾配の移動平均の 指数関数的減衰を合計することにより加速される。 NAdam の学習率を 0.002, β₁ を 0.9, β₂ は 0.999 とする。

Rectified Adam (RAdam) [25] は、学習過程の分 散と運動量の影響に基づいて、適応学習率を自動的 に調整する。高速で安定した最適化アルゴリズムを 使用するために、低い学習率を用いる初期段階にお けるウォームアップという方法も備えている。本研 究では、RAdam の学習率を 0.001、β₁ を 0.9、β₂ は0.999とする。

上述した式において、 g_t , $J(\theta_t)$ 、 η 、 θ_t , γv_{t-1} , G_t RMS, $E[g^2]_t$, \hat{v}_t と \hat{m}_t はそれぞれ,評価関数の勾配, 評価関数, 学習率, 最適化パラメータ, 一期前の勾 配情報を伝える比率. 時間 t までの二乗和勾配の対 角行列, 二乗平均平方根誤差, 時間 t までの移動平 均、二乗勾配の不偏推定量と二乗勾配(平均)の推 定量を意味する。

6. 軽量モデル実験用データセットの作成

6.1 データセット

本研究の計算実験には、Patch-Camelyon benchmark dataset [26], ICIAR2018[27], BreCaHAD [28] 病理写真データセットを使用する。 この3つのデータセットには、合計220587枚の病 理写真があり、写真は正常組織とがん組織に分かれ ている。

表3 データセットの内訳

データセット	写真の 枚数	解像度(px)
Patch-Camelyon (PCAM)	220025	96 × 96
ICIAR2018	400	2048×1536
BreCaHAD	162	1360×1024

px:ピクセル

6.2 CNN モデルのメモリ使用量

上述した深層学習のための CNN モデルは、複数 の層から構成されており、層の内部にも多くの ニューロンが存在している。学習によって損失値を

最小にするには、すべてのニューロンはその重さを確定しなければならないことから、一つの CNN モデルには膨大なパラメータが存在する。

また、CNNが必要とする行列演算には、GPU (Graphical Processing Unit)が適しているとされている。実際に、同じCNNモデルをそれぞれCPUとGPUで学習したところ、モデルの大きさに応じてGPUはCPUより学習が数倍から数百倍速くなる。その一方、CNNを実装するとき、これらのパラメータを格納できるようにメモリを分配しなければならないため、多くのGPUメモリが必要とされている。CNNが消費するメモリ量の計算は以下になる。

メモリ消費量= (ニューロン数×バッチサイズ+重みパラメータ数) × 2 × 4

この式の中の、バッチサイズは一回に投入する入 力画像のデータ量を意味する。また、ニューロン数 と重みパラメータ数は入力データの解像度に関係す る。解像度が高ければ、ニューロン数と重みパラメー タ数も増える。なお、CNN モデルのメモリ消費以 外に、データをモデルに入力すればメモリを消費す るため、括弧内の数値を2倍にし、さらにニュー ロンの重みは単精度浮動小数点数で保存されている ため、さらに4倍のメモリを消費する。

EfficientNet-B6 を使用して、バッチサイズを 64 とする場合のメモリ消費量を試算すると、解像度 96px × 96px の写真では約 10GB のメモリを消費 する。それに対して、解像度 512px × 512px の写真では約 300GB のメモリを消費する。

大規模なデータセンターでもなければ、300GB 以上のGPUメモリを確保することができないため、解像度の高い写真をデータとして直接CNNに 入力することができない。本研究に使われる ICIAR2018とBreCaHADのデータセットも写真 の解像度が高いため、それを学習するには何らかの 処理が必要である。

6.3 病理写真の処理

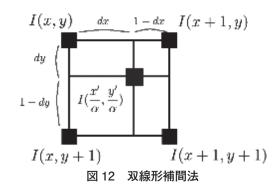
上述したように、解像度の高い病理写真を CNN に学習させる際に、そのままでは直接に入力できないため、前処理が必要である。その処理方法として、クロップ、リサイズと分割後リサイズの 3 つが挙げられる。また、本研究で使われるデータセットの中では、PCAM の病理写真の解像度が 96px × 96px で最も低いため、ほかのデータセットもそれ

に合わせ病理写真を前処理で96px×96pxにする。

クロップ処理は縦横の画素を指定し画像の一部を切り出す操作で、非常に単純な前処理であるため、Teachablemachine [29] などのオンライン深層学習サービスに使用されている。しかしながら、1360px × 1024px から 96px × 96px の領域を切り出して使用するには、画像の 99% 以上を破棄することになるため、病理写真が持つ情報がほとんど失われてしまう恐れがある。よって、この方法は適していないと判断した。このため、本研究はリサイズと分割後リサイズの 2 つの前処理手法を実験し議論する。

6.3.1 リサイズ

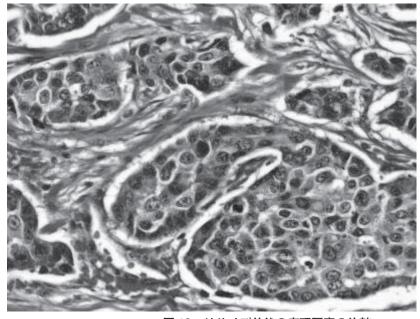
リサイズ処理は、画像を指定した解像度(本研究では 96px × 96px)に変形する処理である。なお、本研究で使用する画像を変形する手法は双線形補間法(バイリニア法)である。双線形補間法では、求めたいピクセルの周辺の 2 × 2 ピクセルの輝度値を参照し、その加重平均を用いて補間する。双線形補間法の詳細は図 12 で説明する。



図で示したように、拡大画像の座標 (x', y') における画素値を求めるには

- ①拡大画像の座標 (x', y') を拡大率 α で割り, $(x'/\alpha, y'/\alpha)$ を求める。
- ②元画像における (x'/a, y'/a) の周囲 4 画素の画素 値 I(x, y), I(x+1, y), I(x, y+1), I(x+1, y+1) を取得する。
- ③周囲 4 画素それぞれと (x'/α, y'/α) との距離を求める。
- ④距離と反比例して重み付け $(0 \sim 1)$ を行う。
- ⑤周囲 4 画素の画素値の加重平均を拡大画像の 座標 (x', y') における画素値。すなわち、I'(x', y')=(1-dx)(1-dy)I(x, y)+dx(1-dy)I(x+1, y)+(1-dx)dyI(x, y+1)+dxdyI(x+1, y+1)

図13は、双線形補間法でリサイズする前後の病



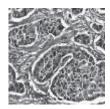


図 13 リサイズ前後の病理写真の比較 (左:リサイズ前(視認性確保のため 1/3 縮小)、右:リサイズ後(等倍))

理写真を示している。高解像度から 96px × 96px にリサイズした病理写真は、細胞の形などの情報は そのまま保持しているのに対し、細部の情報はほと んど失われていると考えられる。

6.3.2 分割後リサイズ

リサイズ処理後に写真細部の情報を保持できない 欠点をカバーするため, 本研究には分割後リサイズ という手法を使用する。

まず一枚の病理写真を複数枚に均等に分割し、そ の分割した写真を $96px \times 96px$ にリサイズする。 図 14 で示したように、分割後リサイズ処理した後 の写真にある細胞の大きさが揃うように、2048px × 1536px の ICIAR2018 の病理写真を 25 等分に, 1360px × 1024px の BreCaHAD の病理写真を 4 等分にすることに決めた。

なお、分割した写真は、6.3.1と同じ双線形補間 法を利用してリサイズする。分割後リサイズした写 真はリサイズした写真と比べて全体的な縮小率が低 いため、細部の情報を比較的多く保持することがで き、CNN による深層学習に有利だと考えられる。

以後、このように処理して得られたデータセット を軽量化データという。

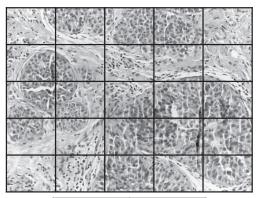
6.4 実験用データセット

6.3.1 及び 6.3.2 で述べた手法でリサイズ或いは 分割してリサイズした ICIAR2018. BreCaHAD の

病理写真と PCAM のデータを用いて本研究の学習 用データを作成する。

6.4.1 不均衡データの Undersampling

表3のデータを軽量モデルに変換した後の結果 を図4に示す。なお、軽量化データの病理写真の 解像度はすべて $96px \times 96px$ になる。写真の枚数は、 PCAM では元データセットと変わらないが. ICIAR2018 と BreCaHAD では分割のため増加し ている。



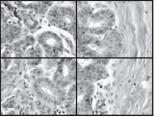


図 14 病理写真の分割法 (上:ICIAR2018 (25 等分), 下:BreCaHAD (4 等分))

つぎに、ICIAR2018、BreCaHADの枚数はPCAMに比べ非常に少ないので、3つのデータセットにあるすべての写真を学習に用いれば、ICIAR2018、BreCaHADデータセットの影響が非常に少ないと考えられる。このため、学習用データを作成するのに、Undersampling処理が必要である。

6.4.2 比較用データセットの作成

リサイズ手法と分割後リサイズ手法で作成したデータの学習精度を比較するため、まずリサイズのみのデータセットを作成する。分割前に最も枚数が少ない BreCaHAD データセットの元写真枚数(162枚)を基準にし、他2データセットの中の写真を162枚ずつ無作為に抽出し、合計486枚の元病理写真を含むデータセットを得た。その後、ICIAR2018と BreCaHAD データセットに含めたすべての写真をリサイズした。このように、合計486枚の病理写真を含むデータセットを作成した。それに対し、分割後リサイズのデータセットは、分割後リサイズのデータセットは、分割後リサイズのデータセットは、分割をリサイズのデータセットは、分割をリサイズを手法で処理したICIAR2018、BreCaHADの写真とPCAMの写真からそれぞれ162枚を無作為に抽出し、データセットを作成した。

表 4 分割後リサイズデータセット内訳

写真の枚数	解像度(px)
220025	96 × 96
10000	96×96
648	96×96
	220025 10000

6.5 リサイズと分割後リサイズした写真の学習比較

6.4.1 で記述した方法で作成した 2 つのデータセットを CNN モデルに入力し、学習の結果を比較する。学習用 CNN モデルは Efficient-B7、活性化関数は Mish、最適化アルゴリズムは RMSProp を使用する。バッチサイズ 64 で 50Epoch の学習をした後、テストデータの認識精度と学習ロスを比較する。

6.6 リサイズと分割後リサイズした写真の学習比較 結果

6.4.2 で記述した方法で作成した 2 つのデータ セットを CNN モデルに入力した学習の結果を表 5 に示す。

リサイズのみで作成されたデータセットの学習

は、分割後リサイズで作成されたデータセットの学習よりロスが若干大きく、認識率は低いことが分かった。このため、解像度の大きい病理写真をCNNで学習する時に、写真をあらかじめ分割しリサイズしたほうが、より良い学習結果につながることが示唆される。

表 5 リサイズと分割後リサイズした写真の学習比較結果

手法	学習ロス	認識率
リサイズ	0.27	0.8729
分割後リサイズ	0.23	0.8933

6.7 学習用データセットの作成とその内訳

6.6 の実験結果を受け、分割後リサイズした ICIAR2018、BreCaHAD データセットの写真を CNN モデル、活性化関数と最適化アルゴリズムの 選択実験のデータとして使用する。このため、写真を分割した後で最も少ない BreCaHAD データセットの写真枚数(648 枚)を基準にし、正常写真とが んの写真が同じ数になるように他 2 データセット の中の写真を 648 枚ずつ無作為に抽出しリサイズ した。このように、合計 1944 枚の病理写真を含む データセットを作成した。

7. 高性能 CNN モデルの探索

7.1 実験環境

実験用データセットの前処理や分割などの操作は、AMD Ryzen 7 1700 Eight-Core Processor @3.00 GHz と NVIDIA GeForce RTX 3090 が配備しているローカルマシンで行った。

すべての実験は Intel Xeon @ 2.20GHz CPU と Tesla P100 GPUを配備している Google Colaboratory [30] で行っている。また、プログラミング言語は Python、ニューラルネットワークライブラリは Tensorflow [31] と Keras [32] を使用している。

また、すべての実験は訓練―検証―テストで交差検証を行うため、データセットは訓練用(70%)、検証用(15%)とテスト用(15%)の3つのサブセットに分けている。画像認識の精度を上げるため、画像のバリエーションを生成する image data generator、学習率を自動的に調整する reduce learning rate on plateau が適用される。バッチのサイズを 64 とし、Epoch 数は 50 とする。

7.2 CNN モデルの選択実験

CNN モデルの各部が画像認識の精度へ与える影響を調べるために、この実験では活性化関数はMish、最適化アルゴリズムは RMSProp を用いる。CNN モデルを変更し、モデルの重みパラメータ数、学習結果(各ニューロンの重み)を保存する H5 ファイルの大きさ、テストデータの認識率と Epoch 時間を比較する。また、CNN モデルの選択実験にて性能を比較する。モデルは MobileNetV2/V3、InceptionResNetV2、EfficientNetB2/B3/B6/B7 とする(表 1)。

7.3 活性化関数の選択実験

活性化関数の画像認識の精度への影響を調べるために、この実験では CNN モデルは EfficientNetB6、最適化アルゴリズムは RAdam を使用する。活性化関数を変更して、テストデータの認識精度と Epoch時間を比較する。また、比較に用いる活性化関数は ReLU、Swish と Mish である。

7.4 最適化アルゴリズムの選択実験

最適化アルゴリズムの画像認識の精度への影響を調べるために、この実験ではCNNモデルはEfficientNetB6、活性化関数はMishを使用する。最適化アルゴリズムを変更して、テストデータの認識精度とEpoch時間を比較する。また、比較する最適化アルゴリズムはSGD、Adagrad、Adadelta、RMSProp、Adam、Adamax、Nadam とRAdamである。

8. CNN モデルによる実験結果と考察

第7章の方針に従って、分割後リサイズによって得られたデータセット(軽量化データ)を用いて CNN モデルによる深層学習の計算実験を行い比較 した。その結果を以下に示す。

8.1 CNN モデル選択実験の結果

計算実験によって、種々の CNN モデルについて 得られた結果を表 6 に示す。なお、この実験では 活性化関数は Mish、最適化アルゴリズムは RMSProp を用いている。

MobileNet 系 CNN モデルは、パラメータと H5 ファイルのサイズが小さいため、計算資源の節約に つながる。一方、テストファイルの認識率は比較的 低く、実用化できるレベルに達していない。

InceptionResNetV2 はパラメータと H5 ファイルのサイズが大きいため、計算資源を多く消費する。 認識率は 0.9107 で実用化できるレベルに達している。 その他、InceptionResNetV2 は同じ規模の EfficientNetB6 と EfficientNetB7 より Epoch 時間が短いため、大量データによる学習の場合に時間が節約できる。

EfficientNet 系 CNN モデルは、そのパラメータと H5 ファイルのサイズによってさらに 2 種類に分けることができる。規模の小さい EfficientNetB2と EfficientNetB3 はパラメータと H5 ファイルのサイズが小さく、計算能力が低いエッジ端末でも実装できる。それに対して、大規模な EfficientNetB6と EfficientNetB7は GPU のあるコンピュータにしか実装できず、Epoch 時間も長くなる一方、テストデータの認識率が高いことが特徴である。

表 6 CNN モデル選択実験の結果

CNN モデル	総重みパラメータ数	H5 ファイルの大きさ	Epoch 時間	認識率
MobileNetV2	2.9M	23MB	7s	0.7423
${\bf Mobile Net V3}$	4.8M	38MB	$7\mathrm{s}$	0.5052
Inception Res Net V2	55M	423MB	6s	0.9107
${\bf Efficient Net B2}$	8M	66MB	6s	0.9313
Efficient Net B3	11M	89MB	6s	0.9313
EfficientNetB6	42M	480MB	11s	0.9347
EfficientNetB7	65M	501MB	12s	0.9381

8.2 活性化関数選択実験の結果

活性化関数選択実験の結果は表7に示す。なお、この実験ではCNNモデルはEfficientNetB6、最適化アルゴリズムはRMSPropを使用している。

ReLU を使用する場合は平均 Epoch 時間が最も 短かった。これは、ReLU が区分線形関数だけであ り、計算するのが容易なためである。

Swish は EfficientNet 系 CNN モデルのデフォルトの活性化関数である。安定的で良好な表現で様々な研究に用いられたが、本研究では Mish よりやや精度が落ちることが分かった。

Mish は ReLU より精度が高いと提唱者が唱えており、本研究では ReLU と Swish よりわずかに認識率が高かった。また、Mish は複雑な定義式を持つため、それを用いる場合の Epoch 時間が最も長かった。

表 7 活性化関数選択実験の結果

活性化関数	Epoch 時間	認識率
ReLU	9s	0.9107
Swish	10s	0.9141
Mish	11s	0.9347

8.3 最適化アルゴリズム選択実験の結果

最適化アルゴリズムが CNN の画像認識の精度へ与える影響を調べるために、この実験では CNN モデルは EfficientNetB6、活性化関数は Mish を使用している。比較する最適化アルゴリズムは SGD、Adagrad、Adadelta、RMSProp、Adam、Adamax、Nadam と RAdam で、それぞれの結果は表 8 に示す。

Epoch 時間は、RAdam を使用する場合に最も長かったが、どの最適化アルゴリズムを使用してもそれほど変わらないことが分かった。

また、認識率の面では、RMSPropが最も高かったが、RMSProp以降に改良された最適化アルゴリズムは基本的に高い認識率が得られることが分かった。

8.4 結論

分割後リサイズの手法で作成された病理写真データを用い,高性能 CNN モデルを探索した結果,

CNN の構造: EfficientNetB7.

活性化関数: Mish.

最適化アルゴリズム: RMSProp

表 8 最適化アルゴリズムの CNN の画像認識精度への影響

最適化アルゴリズム	Epoch 時間	認識率
SGD	10s	0.6735
Adagrad	10s	0.6838
Adadelta	10s	0.5292
RMSProp	11s	0.9347
Adam	10s	0.9072
Adamax	10s	0.9175
Nadam	11s	0.8935
RAdam	12s	0.9244

の組み合わせが最も有望であることが分かった。このモデルを用いれば、0.9381の識別率を得ることができた。一方、EfficientNetB7は大規模なCNNモデルであるため、一般のコンピュータに実装することが難しいと考えられ、医療現場では、

CNN の構造: EfficientNetB2.

活性化関数: Mish,

最適化アルゴリズム: RMSProp

の組み合わせを勧めることができる。このモデルを 用いれば、0.9313の識別率を得ることができた。 また、学習データを変更したところ、平均で約 0.9310の識別率を確認した。

表 9 識別結果の混同行列

予測クラス

		正常	がん
実際点	正常	145	2
クラス	がん	18	126

さらに、表9の混同行列で示した通り、本モデルにおいて正常写真の再現率(Recall、正確に判断したもの÷判断したいデータ)は0.98に達した。

9. 結言

本研究は、日本における専門病理医が圧倒的に不足していることから、がんの病理診断にかかる時間が長いという課題を解決すべく、CNNを用いた深層学習による診断を実験し、実用化できる可能性を検証したものである。

そのために、まず CNN による深層学習において、 解像度の大きい病理写真を直接入力できないという 問題点に対し、写真を分割した後リサイズするとい う解決策(軽量化)を提案した。これにより、病院 で使用される一般のコンピュータでも CNN で病理 写真を診断できるようになる。

また、高性能 CNN モデルを探索する計算実験の結果、8.4 に示したモデルを推奨できたが、この CNN モデルは、重みパラメータ数が比較的少ない(その結果、計算時間も短い)にもかかわらず画像識別の精度が高いことが判明した。したがって、次のような方式が実用的と考えられる。すなわち、現実の臨床診断において、推奨 CNN モデルを診断用コンピュータに実装し学習した後、診断したい病理写真を分割し、CNN に識別させ、分割した写真の中に1枚でもがんと判断したものがある場合は元写真をがんと判断するといったルールで、CNN が正常と判断した元写真以外の写真のみを病理医に精査させるような運用方法である。本研究の結論として、そのような方式を提案したい。

10. 謝辞

本研究の実施及び論文の作成に当たり, 懇切丁寧 な御指導を賜りました京都情報大学院大学応用情報 技術研究科 茨木俊秀教授及び望月バドル助教に御 礼申し上げます。

また、STマイクロエレクトロニクス社 ファイロザ アミラビンティ ハムザー氏には、数々の議論を通して数多くアドバイスをいただきました。この場を借りて、お礼を申し上げます。

参考文献

- [1] 国立がん研究センター, "最新がん統計", https://ganjoho.jp/reg_stat/statistics/stat/summary.html
- [2] 厚生労働省, "病理専門医", https://www.mhlw.go.jp/stf/shingi/2r9852000002asiu-att/2r9852000002ass5.pdf
- [3] Y. LeCun, Y. Bengio, G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.
- [4] I. Goodfellow et al, "Convolutional Networs," in Deep Learning, MIT Press, 2016, pp. 330-372.
- [5] C. Szegedy, Wei Liu et al, "Going Deeper with Convolutions", arXiv:1409.4842, 2014.
- [6] K. He, X. Zhang et al, "Deep Residual Learning for Image Recognition", arXiv:1512.03385, 2015.
- [7] Andrew G. Howard, et al. "Mobilenets: Efficient convolutional neural networks for mobile vision

- applications." arXiv:1704.04861, 2017.
- [8] M. Sandler, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
- [9] A. Howard, et al. "Searching for mobilenetv3." arXiv:1905.02244, 2019.
- [10] M. Tan, Quoc V. Le., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.," arXiv:1905.11946, 2019.
- [11] R. H. R. Hahnloser et al, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," Nature, vol. 405, pp. 947-951, 2000.
- [12] C. E. Nwankpa et al, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," arXiv:1811.03378v1, 2018.
- [13] P. Ramachandran, B. Zoph, Q. V. Le, "Searching for Activation Functions," arXiv:1710:05941v2, 2017.
- [14] D. Misra, "Mish: A Self Regularized Non-Monotonic Neural Activation Function," arXiv:1908.08681v2, 2019.
- [15] I. Goodfellow et al, "Numerical Computation," in Deep Learning, MIT Press, 2016, pp. 80-97.
- [16] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," The Annals of Mathematical Statistics, vol. 23, no. 3, pp. 462-466, 1952
- [17] I. Goodfellow et al, "Optimization for Training Deep Models," in Deep Learning, MIT Press, 2016, pp. 274-329
- [18] J. Duchi, E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," Journal of Machine Learning, vol. 12, pp. 2121-2159, 2011.
- [19] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747v2, 2017.
- [20] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv:1212.5701v1, 2012.
- [21] G. Hinton, "rmsprop: Divide the gradient by a running average of its recent magnitude", http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
- [22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in Int. Conference on Learning Representations, 2015.
- [23] T. Dozat, "Incorporating Nesterov Momentum into Adam," in Int. Conf. on Learning Representation Workshop, 2016.
- [24] L. Liu et al, "On the variance of the adaptive learning rate and beyond," arXiv:1908.03265v1, 2019.
- [25] Kaggle, "Histopathologic-cancer-detection", https://www.kaggle.com/c/histopathologic-cancer-detection
- [26] G. Aresta, T. Araújo et al, "BACH: Grand challenge on breast cancer histology images", Medical Image Analysis, Volume 56, 122-139, 2019

- [27] A. Aksac, Douglas J. Demetrick et al, "BreCaHAD: a dataset for breast cancer histopathological annotation and diagnosis", BMC Research Notes volume 12, Article number: 82, 2019
- [28] "Teachable Machine", https://teachablemachine. withgoogle.com/
- [29] "Google Colab", https://colab.research.google.com/
- [30] "Tensorflow", https://www.tensorflow.org/
- [31] "Keras", https://keras.io/

▶著者紹介

宜蒙 孫 Yimeng Sun

京都情報大学院大学 助教 同志社大学生命医科学部卒 理学士 京都情報大学院大学 情報技術研究科修了 情報技術修士 (専門職)

コラム

原則反時計回り

作花 一志

太陽系には水金地火木土天海の8個の惑星があるが、その他に小惑星、彗星などサイズが10km 足らずの 小天体が無数にある。このうち小惑星は軌道が確定しているものだけでも約80万個に達する。小さくて暗 くて観測限界以下のものはいくつあるか解らない。

これらはほとんど太陽の周りを左回りすなわち反時計方向に公転している。小惑星の軌道面と地球の軌道 面のなす角度を軌道傾斜角というが、その値が90度以下なら反時計回り、90度以上なら時計回りである。 その分布を調べてみると 30 度以下が 99.2%で、150 度以上はわずか 0.004%に過ぎない。すなわち逆行小 惑星は例外と言える。

角度範囲	個数
0~30	788469
30 ~ 60	6162
60 ∼ 90	127
90 ~ 120	38
120 ~ 150	29
150 ~ 180	34

ところで木星土星などは多数の衛星を従えているが、その中 に母惑星の自転方向と逆方向に公転しているもの(逆行衛星) があり、それらは母惑星から遠く離れた軌道上を回っている。 逆行小惑星も海王星の彼方, 太陽からずいぶん遠い軌道上を公 転しているようだ。

しかしこの現象は天界に限ったことではない。アスリートの世界は左回りである。400m 走などのトラッ ク競技、スピードスケート、野球で打者の走る方向など皆反時計である。フィギュアスケートのスピンでは たまに右回転もあるが、原則は左回転。プレイヤーが左利きだから、競技場が南半球だからといって例外は ない。人間社会も星の社会も同じ法則に従っているのだろうか?