

# 顕微鏡画像からの線虫の健康寿命測定システムの開発

松尾 直哉<sup>1</sup> 上瀧 剛<sup>2</sup> 福島 友太郎<sup>3</sup> 首藤 剛<sup>4</sup>

**概要：**線虫とは、分類学上線形動物門に属する生物の総称である。近年、線虫の一種である *C.elegans* が、新薬開発や寿命研究に活用されている。線虫には死ぬ前に動いてはいるがその場に留まっている期間（不動期）があり、寿命と不動期を測定することで、線虫の健康寿命を特定できる。本稿では、テンプレートマッチング、インスタンス・セグメンテーション、画像分類の技術を用いて、線虫の健康寿命を自動で測定するシステムを提案する。

## Development of a system for measuring healthy life span of *C.elegans* from microscopic images

### 1. はじめに

線虫とは、分類学上線形動物門に属する生物の総称である。近年、線虫の一種である *C.elegans*(図 1) が、新薬開発や寿命研究に活用されてきている [1]。 *C. elegans* は全長約 1mm の小さな生物でありながら、ヒトと同様に、神経・筋肉・消化管などの臓器を持つ。容易に飼育が可能で、寿命は 20~40 日とライフサイクルが非常に短い。この特徴から、新薬開発や寿命研究にマウスやサルの前段階のモデル実験動物として線虫の利用が期待されている。

一方、近年では寿命に加えて「健康寿命」に注目が集まっている [2]。健康寿命とは、WHO(世界保健機関) が 2000 年に提唱した概念で、平均寿命から日常的に医療・介護が必要となる期間を除いた期間を指す。現在、健康寿命の研究においては線虫が用いられている。すなわち、様々な環境・実験条件下での線虫の健康寿命を計測することでその要因を明らかにすることができる。ここで線虫の生死は動いているか否かで確認する。また、線虫には死ぬ前に、動いてはいるがその場に留まっている不動期と呼ばれる期間がある [3]。したがって、寿命と不動期を測定することで健康寿命を測定することができる。現在、線虫の寿命と健康寿命は以下の手順でマニュアルで行われている (図 2)。



図 1: *C.elegans*

- (1) 線虫を約 100 匹飼育したシャーレを、12 時間毎に 30 日間撮影 (合計 50 枚の画像)。
- (2) ある時間の画像を、その 12 時間前後の画像 (計 3 枚) と比較し、線虫 1 匹毎に、(A) 動いている線虫 (健康線虫)、(B) その場に留まっている線虫 (不動線虫)、(C) 全く動いていない線虫 (死亡線虫) のどの状態かを判定。
- (3) 撮影画像全てに (2) を行い、寿命と健康寿命を検出。

しかしながら、これらの一連の手順は膨大な労力と時間を要する。そこで、本研究では画像認識技術を用いて線虫の解析作業を自動化し、研究作業を効率化することを目的としている。

### 2. 関連研究

#### 2.1 物体認識手法

本研究に用いる、機械学習による物体認識手法について説明する。物体認識手法は、出力に応じて以下の 4 種類に分かれている [4]。

<sup>1</sup> 熊本大学大学院自然科学教育部

<sup>2</sup> 熊本大学大学院先端科学研究部

<sup>3</sup> 熊本大学大学院薬学教育部

<sup>4</sup> 熊本大学大学院生命科学研究部

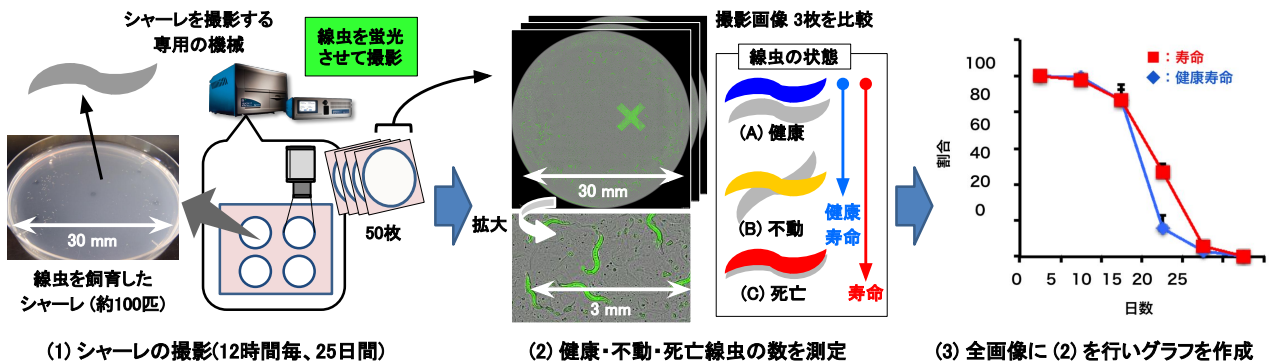


図 2: 線虫の健康寿命の測定手順

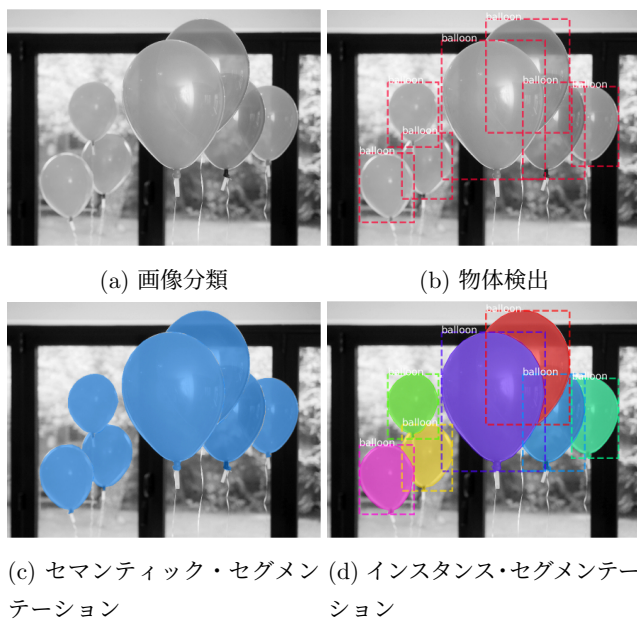


図 3: 物体認識手法 [4]

(1) 画像分類

画像内の物体の種類(クラス)を識別する。物体の位置は検出できない。

(2) 物体検出

画像内の物体の位置とクラスを検出する。物体の位置は、バウンディングボックスと呼ばれる長方形で出力する。複数の同一物体を1つずつ検出できる。

(3) セマンティック・セグメンテーション

画像上の全ピクセルをクラスに分類し、物体ごとに領域を分割する。同クラス間で重なりがある場合は、同クラスの領域として認識するため、物体ごとの認識・カウントはできない。

(4) インスタンス・セグメンテーション

画像内の物体の位置を検出し、領域分割も行う。同クラス間で重なりがある場合でも、物体ごとに認識・カウントできる。

図 3 に上記 (1)~(4) の概略を示す [4]。本稿で提案する線虫の検出手法は (4) インスタンス・セグメンテーション、線虫の生死判定手法は (1) 画像分類に該当する。

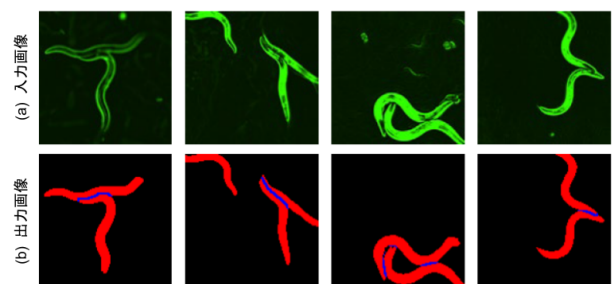


図 4: 線虫のセマンティック・セグメンテーション [5]

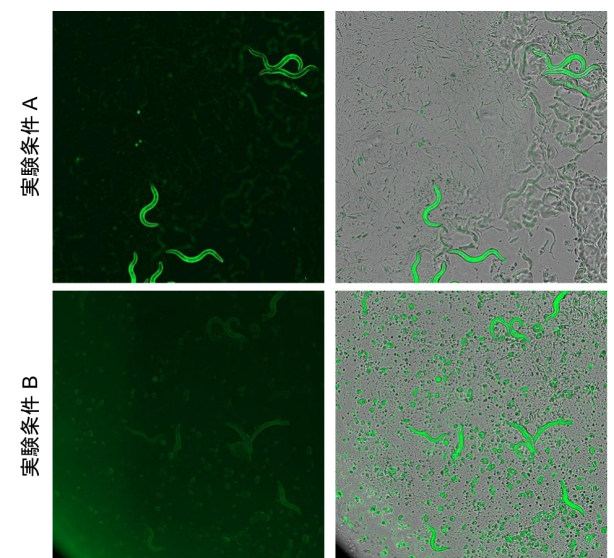


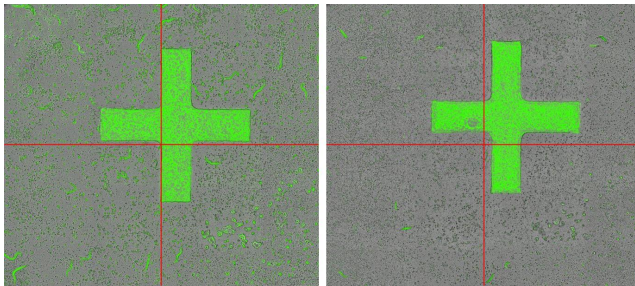
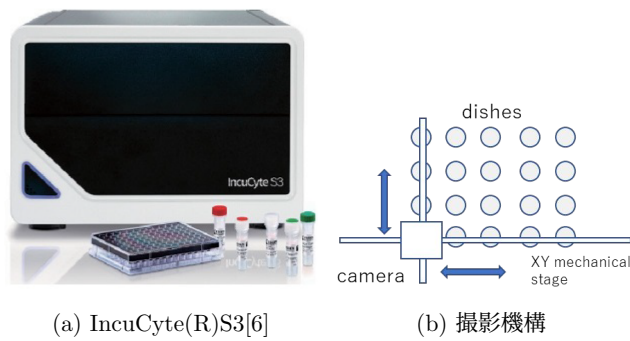
図 5: 撮影画像。(左) 暗視野。(右) 暗視野+明視野。

2.2 健康寿命測定の従来手法

2.2.1 線虫のセマンティック・セグメンテーション

従来手法 [5] では、セマンティック・セグメンテーションを用いて、画像内の線虫を1匹ずつ検出していた。画像には、図 4(a) のように、線虫が2匹以上重なってしまう場合がある。図 4(b) のように、背景(黒)、線虫(赤)、線虫の重なり部分(青)の3クラス分類をすることで境界線を設け、重なった線虫を検出していた。しかし、重なり部分(青)を正確に認識できない場合は分離することができなかった。本研究では、インスタンス・セグメンテーションを用いた、重なった線虫もロバストに検出する手法を提案する。





(c) 撮影画像のずれ (左: 0 フレーム目, 右: 14 フレーム目)

図 6: IncuCyte(R)S3[6]

また、解析には蛍光測定法を用いて線虫を蛍光させた暗視野画像 (図 5) を用いていた。しかし、実験条件によっては、暗視野画像は図 5 の条件 B のように、線虫が鮮明に蛍光できない場合があった。そこで本研究では、実験条件に柔軟に対応できるように、暗視野画像に代わり、暗視野と明視野画像を重ね合わせた画像を解析に用いる。

### 2.2.2 輪郭の形状による線虫の生死判定

従来手法では、以下の手順で線虫の生死判定を行う。

#### (1) 座標判定

あるフレームと一つ前のフレーム (12 時間前) の線虫の重心座標を比較し、座標が近い線虫を探索する。

#### (2) 形状判定

座標が近い 2 匹の線虫に対し、Hu モーメント [7] を用いて、輪郭の形状の類似度を比較する。類似度が閾値よりも高い場合、不動線虫と判定する。また、不動期判定された位置の線虫がその後も不動期判定された場合、48 時間後以降は死亡と判定する。

本研究では、不動線虫と死亡線虫をより正確に判別し、個体毎に不動期を測定するため、深層学習による画像分類を用いた生死判定手法を提案する。

## 3. 提案手法

### 3.1 撮影画像の位置補正

#### 3.1.1 線虫の撮影

線虫の撮影には全自動ハイスループット生細胞解析システム IncuCyte(R)S3 (エッセンバイオサイエンス社) を用いている [5]。図 6(a) に IncuCyte(R)S3 を示す。IncuCyte(R)S3 は細胞実験において短時間に多数の化合物を評価して、新規化合物を迅速に発見する高速大量スクリー

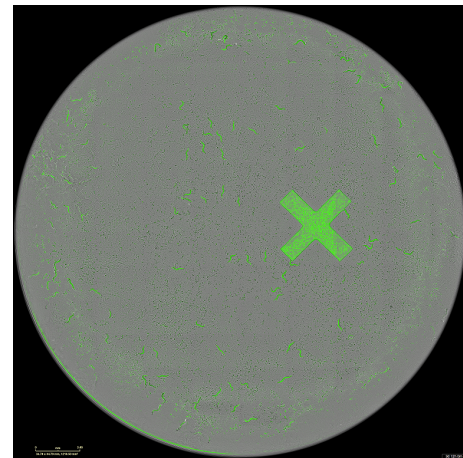


図 7: 撮影画像 (3080×3080pix). 位置補正のため、十字の目印を付けている。

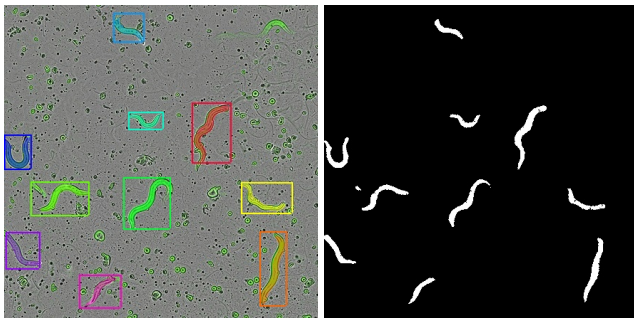
ニング (HTS) が可能であり、引用論文数 1200 を超える導入実績がある装置である [6]。線虫の入ったシャーレ (直径 3cm) を真上から撮影する構造になっており、一定の時間間隔で撮影して、画像を取得することが出来る。また、IncuCyte(R)S3 は細胞蛍光観察を行うことが可能で、蛍光測定法を用いることにより、線虫の像を鮮明に撮影することが出来る。今回の解析には、12 時間ごとに撮影した線虫の時系列画像 50 枚 (25 日分) を使用した。

IncuCyte(R)S3 を用いて撮影した画像を図 7 に示す。撮影画像のサイズは 3080x3080 ピクセルである。線虫の画像上の特徴は、線虫領域が強く蛍光しており、細長い線状の形をしている。また線虫はサインカーブ状の 2~3 個の山を持つ形をしている。そして線虫同士が互いに重なることもある。また、線虫の成長度合いにより、太さや長さが異なる。

#### 3.1.2 撮影画像の位置ずれの補正

IncuCyte(R)S3 は、図 6(b) のように、一度に多数のシャーレを使って実験が可能であり、1 つのカメラを機械ステージを使って移動撮影することで、多数のシャーレを撮影している。しかし、IncuCyte(R)S3 も完璧な装置ではないため、図 6(c) のように撮影する際にステージ誤差が生じる。例えば 1mm のずれでも、撮影画像上では数十ピクセルのずれとなるため、後の生死判定で画像上の位置情報を使用する際に誤判定の原因となる。

これを防ぐために、全撮影画像のシャーレの位置を合わせる前処理を行う。図 7 の撮影画像のように、シャーレに十字の「目印」を付けておき、この目印をテンプレートマッチングを用いて揃える方法で実現する。テンプレートマッチングとは、テンプレート画像を被検出画像上でスライドし、テンプレート画像と類似する領域を探索し、物体を認識する手法である。類似度の計算には、明るさの変化に強いゼロ平均正規化相関 (ZNCC)[8] を用いる。



(a) 検出結果 (可視化) (b) 出力画像

図 8: 線虫のインスタンス・セグメンテーション

### 3.2 線虫の検出

#### 3.2.1 インスタンス・セグメンテーションによる重なった線虫の検出

従来のセマンティック・セグメンテーションによる線虫検出手法では、重なった線虫を検出できない場合があった(2.2.1節)。そこで、インスタンス・セグメンテーションを用いて、画像内の線虫を1匹ずつ検出することで、重なった線虫を検出する手法を提案する。撮影画像は $3080 \times 3080$ pixelと大きいため、1枚を $7 \times 7 \equiv 49$ 枚に分割した $440 \times 440$ pixelの画像を入力画像とする。インスタンス・セグメンテーションを実行する深層学習モデルには、Mask R-CNN[9](3.2.2節)を用いている。セグメンテーションによって検出された線虫は、図8のように白黒の2値画像で出力し、次の生死判定に使用する。

#### 3.2.2 深層学習モデル

インスタンス・セグメンテーションを行う深層学習モデルとして構築したMask R-CNNの構造を図9に示す。まず、Backboneは、複数の畳み込み層から構成されるネットワークで、入力画像の特徴を抽出する役割を担う。101層のResNet(Residual Network)[10]で構成され、入力画像に対する特徴マップを出力する。次に、RPN(Region Proposal Network)[12]は、特徴マップを用いて物体が存在する領域(バウンディングボックス)の候補を選定する役割を担う。Backboneは非常に深いネットワークのため、入力画像が粗くなる。そこでFPN(Feature Pyramid Network)[11]にて、入力画像をアップサンプリングすることで解像度を回復する。最後に、HeadではRPNで得られた領域候補を切り出し、Fully Connected Layerにてクラス識別とバウンディングボックスの最終決定を行い、Mask branchにてバウンディングボックス内のセグメンテーションが行われる。

#### 3.2.3 データセット

学習には、 $3080 \times 3080$ pixelの撮影画像を $7 \times 7 \equiv 49$ 枚に分割した $440 \times 440$ pixelの画像400枚を用いて作成したデータセットを用いる。学習データが少ないため、Data augmentationとして、学習前に入力画像に対して、以下の画像処理をランダムに0~3回行っている。

- (a) 左右反転
- (b) 上下反転
- (c) 回転(30° 毎)
- (d) スケール変更(縦横比)
- (e) 画素値変更(画像に-40~40のランダムな値を追加)
- (f) 明るさ変更(暗く、または明るくする)
- (g) ガウシアンフィルタ(ぼかす)

### 3.3 線虫の生死判定

#### 3.3.1 画像分類による線虫の生死判定

線虫を1匹ずつ検出後、判定する撮影画像と12時間前後の画像の3画像の同じ部分を比較し、「健康」「不動」「死亡」のどの状態か判別する。具体的には、前節に記述した2値画像3枚を合成し、図10のような合成画像を画像分類で、どの状態かを判定する。

#### 3.3.2 深層学習モデル

画像分類にはCNNを用いているResNet[10]の改良モデルResNet-RS[14]、自然言語処理の分野でSoTAを達成したTransformer[15]に用いられるSelf-Attentionを画像分類に用いているVision Transformer(ViT)[16]、Self-Attentionを単純な多層パーセプトロン(MLP)に置き換え構成されたMLP-Mixer[17]の3種類の深層学習モデルで学習し、精度を比較する。

また、学習の際の最適化手法には、Adam[18]にWarmupと呼ばれる最初に小さい学習率で学習し、通常の学習率で学習させていく手法を取り入れ、収束を安定させ、高速化させたRAdam[19]を用いた。

#### 3.3.3 データセット

学習には、セグメンテーション結果の合成画像を用いる。その際に、不動状態の線虫が少なかったため、Data augmentationとして、2値画像を回転、平行移動し、擬似的に不動状態の画像を作成し、データを増やしている。そして、健康、死亡状態も含めた全画像に、反転、回転、スケール変更を行い、データ数を7700枚まで拡張し、たデータセットを作成した。

### 3.4 個体毎の不動期判定

最後に線虫一匹ごとの不動期を計測する。具体的には、あるフレームで「不動」「死亡」と線虫を判定した場合、そのフレーム以降の画像からその線虫の重心と近い位置にいる線虫を追跡する。そして、全フレームで健康線虫の数、健康線虫+不動線虫の数をそれぞれ計測し、健康寿命と寿命のグラフを出力する。また、自動で判定が難しい線虫も目視で確認できるように、図17のような解析結果の表も出力できるようになっている。



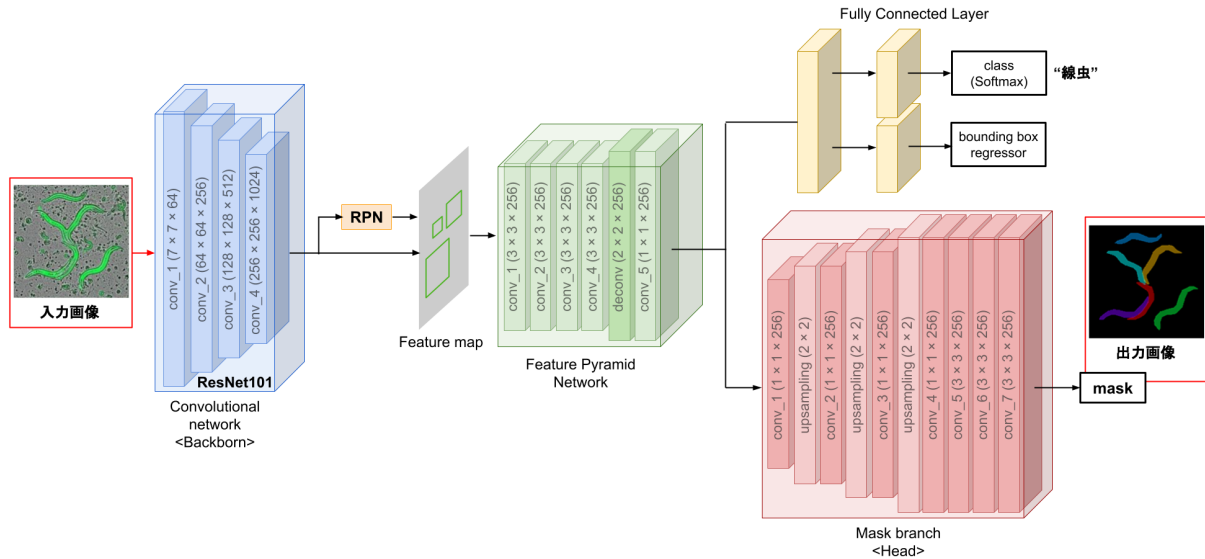


図 9: Mask R-CNN のアーキテクチャ [9]. 画像特徴を抽出後, RPN で物体候補領域を取得し, セグメンテーションを行う。

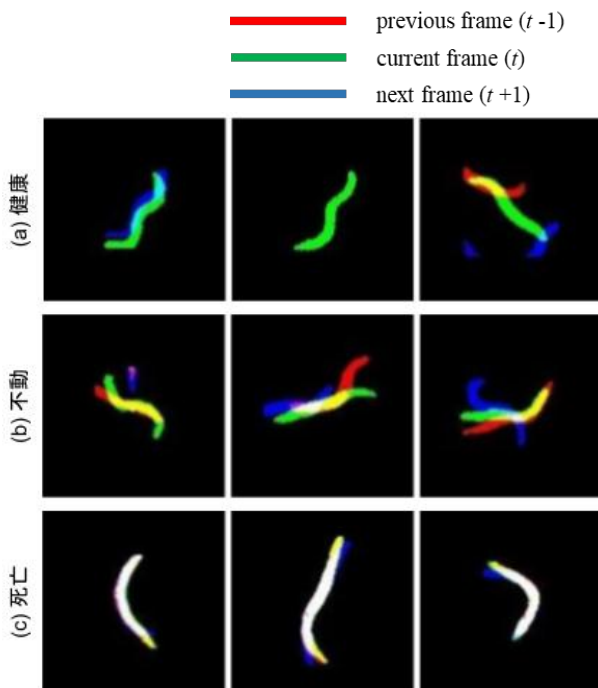


図 10: 合成画像. 画像の 3 クラス分類で生死判定を行う。

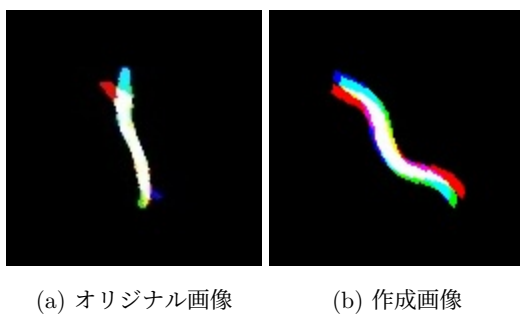


図 11: 「不動」クラスの学習画像

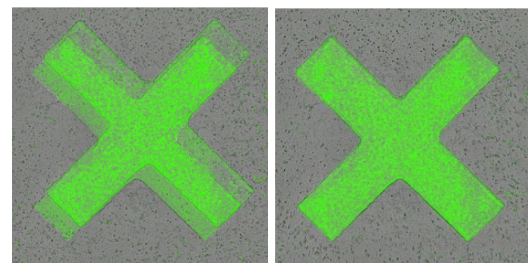


図 12: 検出結果. テンプレートマッチングを用いて, 目印を正確に合わせている。

表 1: セグメンテーションの学習条件

学習回数	300 epochs
バッチサイズ	1
学習率	0.001
最適化手法	SGD[13]

## 4. 評価実験

### 4.1 シャーレの位置補正結果

位置補正を行う前後の比較結果を図 12 に示す. 補正前は目印の位置がズれているが, 補正後は目印の位置を揃えることができています。

### 4.2 線虫の検出結果

#### 4.2.1 学習条件

Tensorow, Keras を用いて Mask R-CNN を実装し, 3.2.2 節で述べたデータセットを表 1 の条件で学習を行った. GPU には, NVIDIA(R) GeForce RTX 2080ti を使用した。

表 2: 線虫の検出精度

検出方法		再現率	適合率	F 値
セグメンテーション	入力画像	[%]	[%]	
セマンティック	暗視野	75.3	99.1	0.856
セマンティック	暗視野+明視野	89.0	98.6	0.935
インスタンス	暗視野	76.6	<b>99.2</b>	0.864
インスタンス	暗視野+明視野	<b>92.2</b>	98.6	<b>0.953</b>

表 3: 重なった線虫の検出精度

検出方法		検出精度
セグメンテーション	入力画像	[%]
セマンティック	暗視野	34.6
セマンティック	暗視野	42.9
インスタンス	暗視野	42.9
インスタンス	暗視野+明視野	<b>60.7</b>

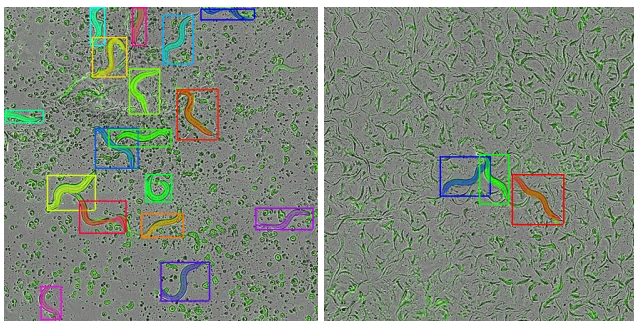


図 13: 検出成功例

#### 4.2.2 定量評価

合計 154 匹の線虫が写ったテストデータ 50 枚を用いて、線虫の検出精度を検証した。提案手法である暗視野+明視野画像にインスタンス・セグメンテーションを行う場合と、セマンティック・セグメンテーションを行う場合、暗視野画像を用いる場合との精度の比較結果を表 2 に示す。また、テストデータには 28 匹の線虫が重なり合っており、重なった線虫のみの検出精度の比較結果を表 4 に示す。

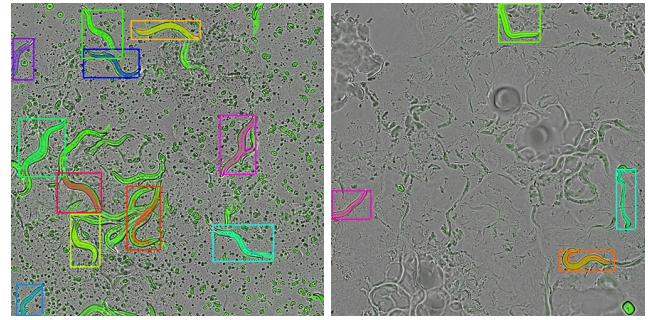
表 2 より、暗視野+明視野画像をインスタンス・セグメンテーションした場合に再現率が最も高くなっている。適合率は暗視野画像を用いた場合の方が僅かに高い結果となった。しかし、誤検出の数は、暗視野画像で 1 つ、暗視野+明視野画像で 3 つと、非常に少なかった。

#### 4.2.3 定性評価

図 13 と図 14 にセグメンテーション結果の一部を示す。図 13 のように、線虫が多数写っている場合や線虫同士が重なっている場合でも検出可能である。しかし、図 14(a) のように、複雑に重なり合った線虫を検出できない場合や、(b) のように線虫以外の物体 ((b) の水色の検出結果はホコリ) を線虫と誤検出してしまふ場合も僅かに見られた。

#### 4.2.4 考察

暗視野画像を用いた場合の方は、適合率は僅かに高いが、



(a) 未検出

(b) 誤検出

図 14: 検出失敗例

表 4: 画像分類の学習条件

使用機器	NVIDIA(R) GeForce RTX 2080ti
モデル	ResNet-RS(CNN)[14], ViT(Attention)[16], MLP-Mixer(MLP)[17]
エポック数	100
バッチサイズ	16
初期学習率	0.001
最適化手法	RAdam[19]

表 5: 生死判定の比較結果

判定方法	再現率 [%]	適合率 [%]	F 値
Hu moment(形状判定)	50.7	51.5	0.511
ResNet-RS(画像分類 1)	86.3	86.5	0.864
ViT(画像分類 2)	91.7	92.6	0.921
MLP-Mixer(画像分類 3)	<b>93.3</b>	<b>93.5</b>	<b>0.934</b>

再現率は暗視野+明視野画像の方がかなり高くなっている。また、インスタンス・セグメンテーションを用いる場合の方が、線虫全体と重なった線虫に絞った検出結果の両方で、セマンティック・セグメンテーションを用いる場合よりも精度が高くなっている。この結果から、暗視野画像にセマンティック・セグメンテーションを行う従来手法よりも、暗視野+明視野画像にインスタンス・セグメンテーションを行う提案手法の方が、より高い精度で線虫を検出できると考えられる。

### 4.3 線虫の生死判定結果

#### 4.3.1 学習条件

3.3.1 節に記載したデータセットを使って、表 4 の条件で 3 種類の深層学習モデルで学習し、精度を比較した。

#### 4.3.2 定量評価

検証には、「健康」「不動」「死亡」の 3 種類の合成画像をそれぞれ 100 枚ずつの計 300 枚を用いて、従来手法(Hu moment)と比較した。表 5 に、各手法の比較結果を示す。3 つのモデル全てが従来手法よりも精度が高くなっており、MLP-Mixer(MLP)[17] が最も高い精度となった。



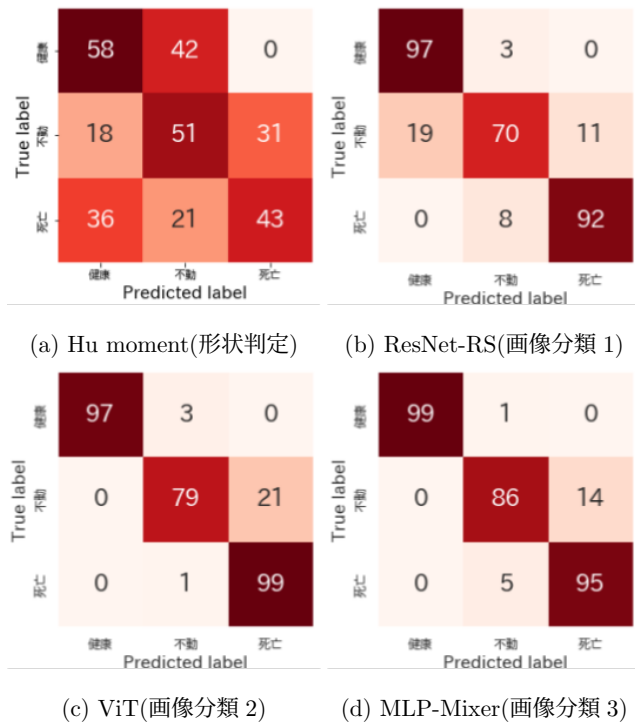


図 15: Confusion Matrix

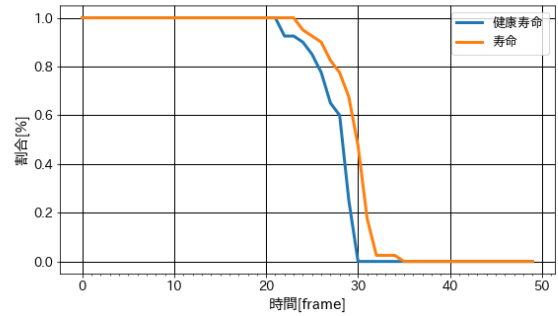
図 15 は、検証結果を Confusion Matrix で示したものである。3つの画像分類モデルで共通して、「不動」だけが精度が低くなっている。原因は、「健康」「死亡」の画像は比較的色などで判別しやすいものが多いのに対し、「不動」の画像には他の2つと似通った画像が多いことが原因だと考えられる。

#### 4.3.3 考察

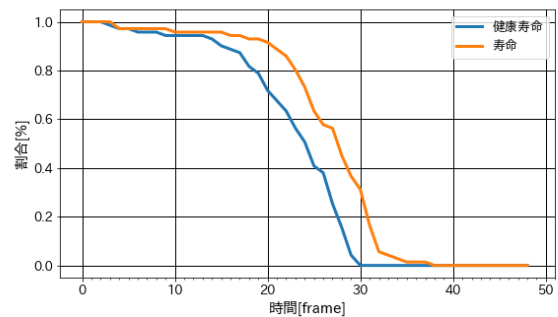
深層学習による画像分類を行う本手法は、従来手法よりも生死判定の精度が大幅に向上し、かなり高い精度で判定が可能になっている。学習モデルの中では、MLP-Mixer[17]が最も高い精度を出している。この要因として、MLP-Mixerの中には、異なるチャンネル間のデータの相互関係を考える channel-mixing MLP と、異なる位置情報間のデータの相互関係を考える token-mixing MLP の2種類の MLP が含まれていることだと考えられる。生死判定に使用する合成画像は、白、黄色の割合で不動、死亡状態を判別できるため、色情報が特に重要だと考えられる。2種類の MLP があることで、色の特徴を特に学習できている MLP-Mixer の精度が最も高くなっていると考えられる。今後は、不動状態の判定精度をより向上させることが課題である。

#### 4.4 健康寿命の解析

開発システムで解析した寿命と健康寿命の結果を目視で解析した場合と比較する。マニュアルで解析したグラフを図 16(a)、開発システムが出力したグラフを図 16(b) に示す。健康寿命は寿命から不動期を引いたものであるため、健康寿命曲線は寿命曲線を少し遅れて追いかける形を示



(a) 目視での解析結果



(b) 開発システムの出力結果

図 16: 寿命と健康寿命のグラフ

す。開発システムを使って、時間経過とともに線虫の生存率が徐々に減っていく寿命モデルを確認できる。しかし、2つを比較すると、前半の健康寿命と寿命の推移が開発システムの方が早く進んでいる。この原因は、セグメンテーションで線虫ではない物体を検出してしまい、不動線虫と判定してしまっていること、健康な線虫を誤って不動状態と判定してしまっていることだと考えられる。今後は、線虫の検出と生死判定の両方の精度を向上させ、目視での解析結果により近づける必要がある。具体的には不動期の測定結果が目視の結果と同程度となる精度を目指す。

## 5. おわりに

本研究では、線虫の健康寿命を顕微鏡画像から自動で測定することを目的として、テンプレートマッチングによる撮影画像の位置補正、インスタンス・セグメンテーションを用いた線虫の検出、画像分類を用いた線虫の生死判定を行う手法を提案した。また、検証実験により従来手法と比較することで、提案手法がより高い精度で線虫の検出と生死判定が可能であることを確認した。

#### 参考文献

- [1] R. Xiao and B. Zhang and Y. Dong and J. Gong and T. Xu and J. Liu and X.Z. S. Xu: *A genetic program promotes c. elegans longevity at cold temperatures via a thermosensitive trp channel*, Cell (2013).
- [2] Yoshio Nakano and Masataka Moriuchi and Tsuyoshi

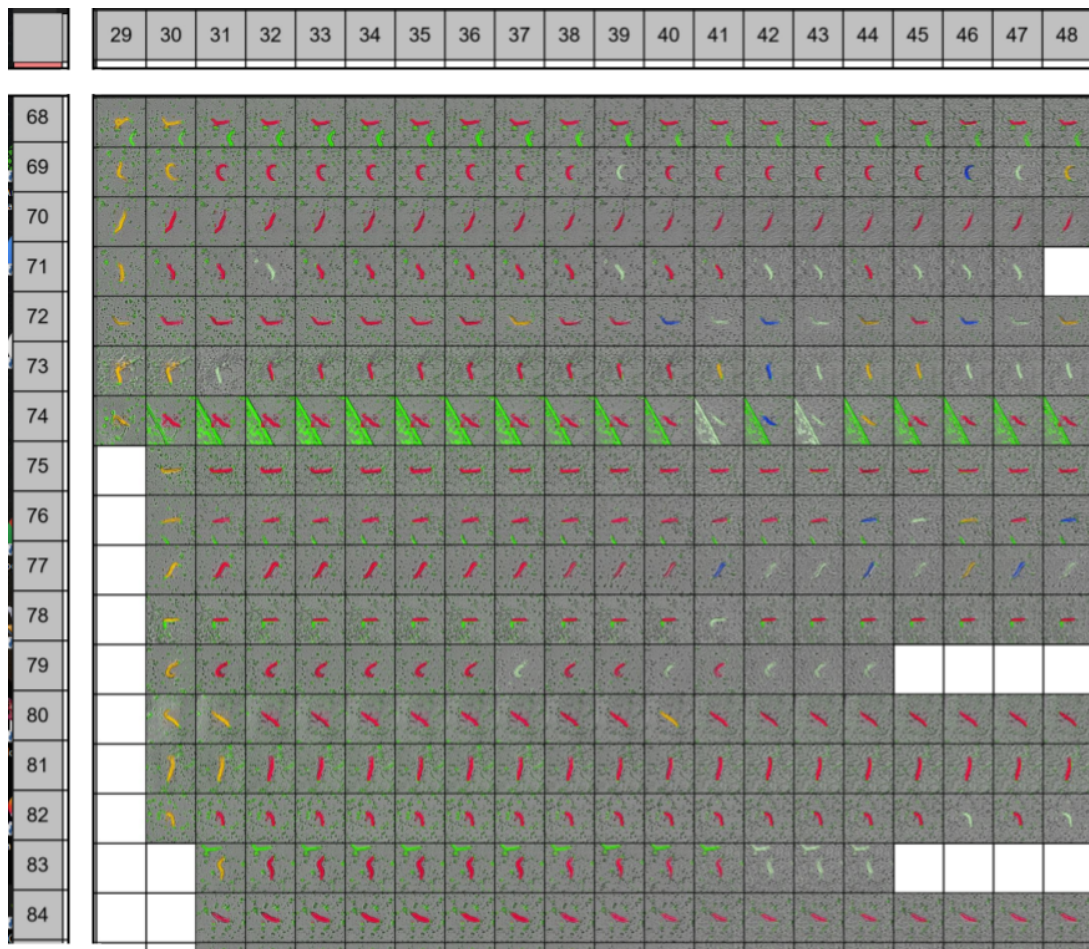


図 17: 個体毎の不動期解析結果 (全体画像は大きすぎるため一部抜粋). 横軸の番号は何フレーム目か, 縦軸の番号は検出した線虫に順に割り振った番号を示している. セグメンテーションで検出した領域の色味を変更している (青: 健康, 黄: 不動, 赤: 死亡, 白: 検出に失敗した画像). この表を出力し, 自動で不動期の判定が難しい線虫も目視で確認できる.

- Shuto and et.al: *Intrapopulation analysis of longitudinal lifespan in *Caenorhabditis elegans* identifies W09D10.4 as a novel AMPK-associated healthspan shortening factor*, Journal of Pharmacological Sciences (2021).
- [3] 森内 将貴: 線虫の寿命制御因子探索に関する研究, Ph.D. thesis (2019).
- [4] Waleed Abdulla: *Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow*, 入手先 <https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46> (2018).
- [5] 林 恭太郎: 深層学習を用いた画像からの線虫の自動判別システムの開発, Master's thesis (2020).
- [6] 全自動ハイスループット生細胞解析システム IncuCyte(R)S3, 入手先 <http://www.wako-chem.co.jp/siyaku/kiki/cell/incucyte/index.htm>.
- [7] M. K. Hu: *Visual Pattern Recognition by Moment Invariants*, IRE Transaction On Information Theory (1962).
- [8] J. P. Lewis: *Fast Normalized Cross-Correlation*, (1995).
- [9] He, Kaiming and Gkioxari, Georgia and Dollár, Piotr and Girshick, Ross: *Mask R-CNN*, Proc. IEEE CVPR (2017).
- [10] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian: *Deep Residual Learning for Image Recognition*, Proc. IEEE CVPR (2015).
- [11] Lin, Tsung-Yi and Dollár, Piotr and Girshick, Ross and He, Kaiming and Hariharan, Bharath and Belongie, Serge: *Feature Pyramid Networks for Object Detection*, Proc. IEEE CVPR (2016).
- [12] Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, Jian: *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Proc. IEEE CVPR (2015).
- [13] Qian Ning: *On the momentum term in gradient descent learning algorithms*, Neural Networks 12.1 (1999).
- [14] Irwan Bello and William Fedus and et.al: *Revisiting ResNets: Improved Training and Scaling Strategies*, Proc. IEEE CVPR (2021).
- [15] Vaswani et al: *Attention is All You Need*, NIPS (2017).
- [16] Alexey Dosovitskiy and Lucas Beyer and et.al: *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, Proc. IEEE CVPR (2020).
- [17] Ilya O. Tolstikhin and Neil Houlsby and et.al: *MLP-Mixer: An all-MLP Architecture for Vision*, Proc. IEEE CVPR (2021).
- [18] Diederik P. Kingma, Jimmy Ba: *Adam: A Method for Stochastic Optimization*, ICLR (2015).
- [19] Liyuan Liu and Haoming Jiang and et.al: *On the Variance of the Adaptive Learning Rate and Beyond*, ICLR (2020).