

追跡ツール「UMATracker」への 深層学習による検出器の導入

椋木研 B4 学籍番号：67170390
比江島 惇

目次

導入

1. 研究内容
2. 研究背景
3. 目的

内容

4. UMATracker
5. UMATrackerの問題点
6. 深層学習による検出器の導入
7. YOLOを用いた小動物追跡
8. 実験
9. 結論

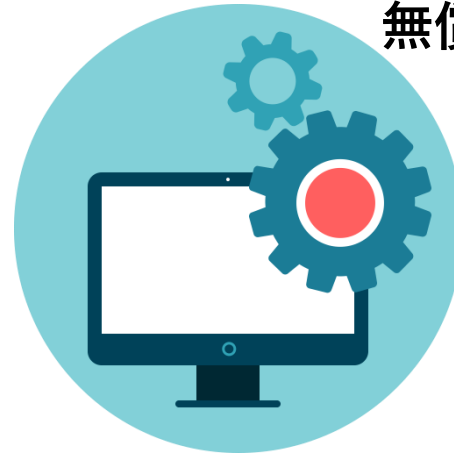
研究内容

従来の物体追跡ツール「UMATracker」に、深層学習による検出器(YOLOによる検出)を導入する。

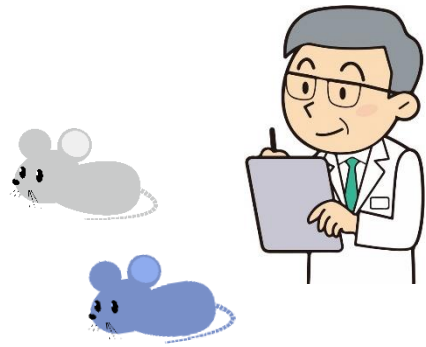
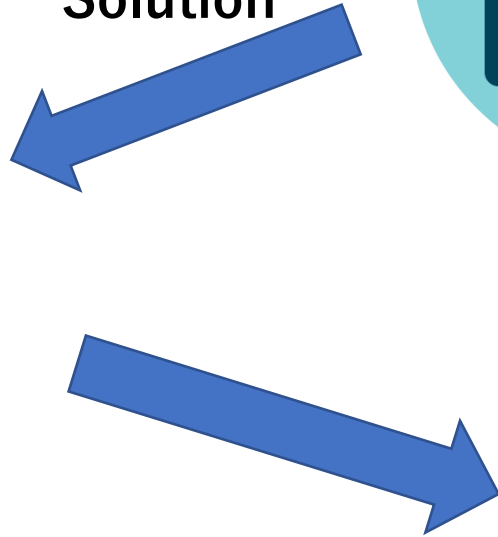
それを、飼育環境下での複数の小動物(マウス)の同時追跡に適用することで、従来のマウス追跡の性能を向上させる。

研究背景

無償の様々な追跡ツール

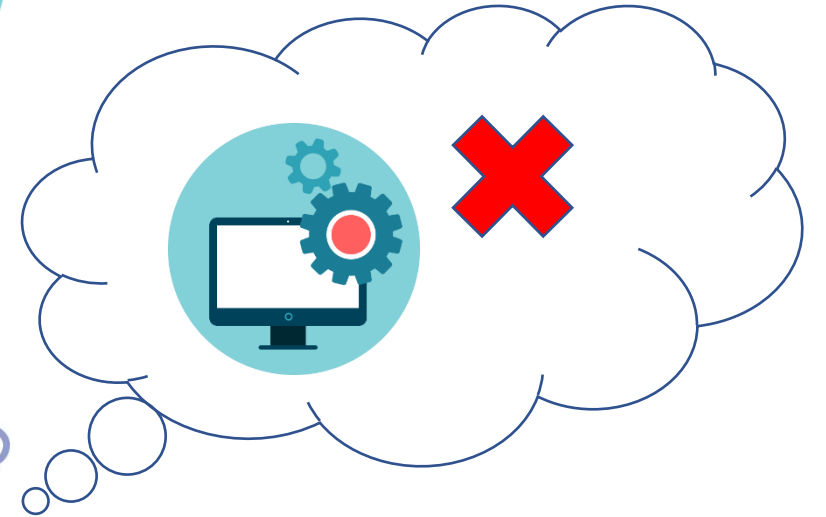


Solution



従来、
マウスの行動観察を
手作業で行っている。

観察者に負担がかかる。



古い技術をベースに
しているため、追跡精度
は良くない

目的

従来の追跡ツールから比較的**ユーザビリティに富んだもの**を選定し、そのツールに**深層学習**を取り入れる。

**深層学習を用いた検出(YOLOの検出)により
従来ツール(UMATracker)の追跡精度を改良**



マウス追跡の性能 向上

UMATracker

小動物の追跡ツール

項目	pathtracker (2019)	MOTHe (2020)	ezTrack (2019)	Tracktor (2019)	UMATracker (2018)
複数体の追跡	×	○	×	○	○
作業の単純さ	○	×	○	○	○
GUIの有無	無	無	無	無	有
検出の精度	良	機械学習の データ強化による	入力動画 による	パラメータ設定 による	パラメータ設定 による
パラメータ設定の 難易度	-	-	-	難しい	易しい
オクルージョン への対応	×	×	×	×	○

GUIの有無、パラメータ設定の難易度、オクルージョンへの対応の項目から、UMATrackerをベースの追跡ツールに決定した。

UMATracker

GUIがあるため、作業が視覚的にわかりやすい。
3つのプロセスを経て、物体追跡を実現する。

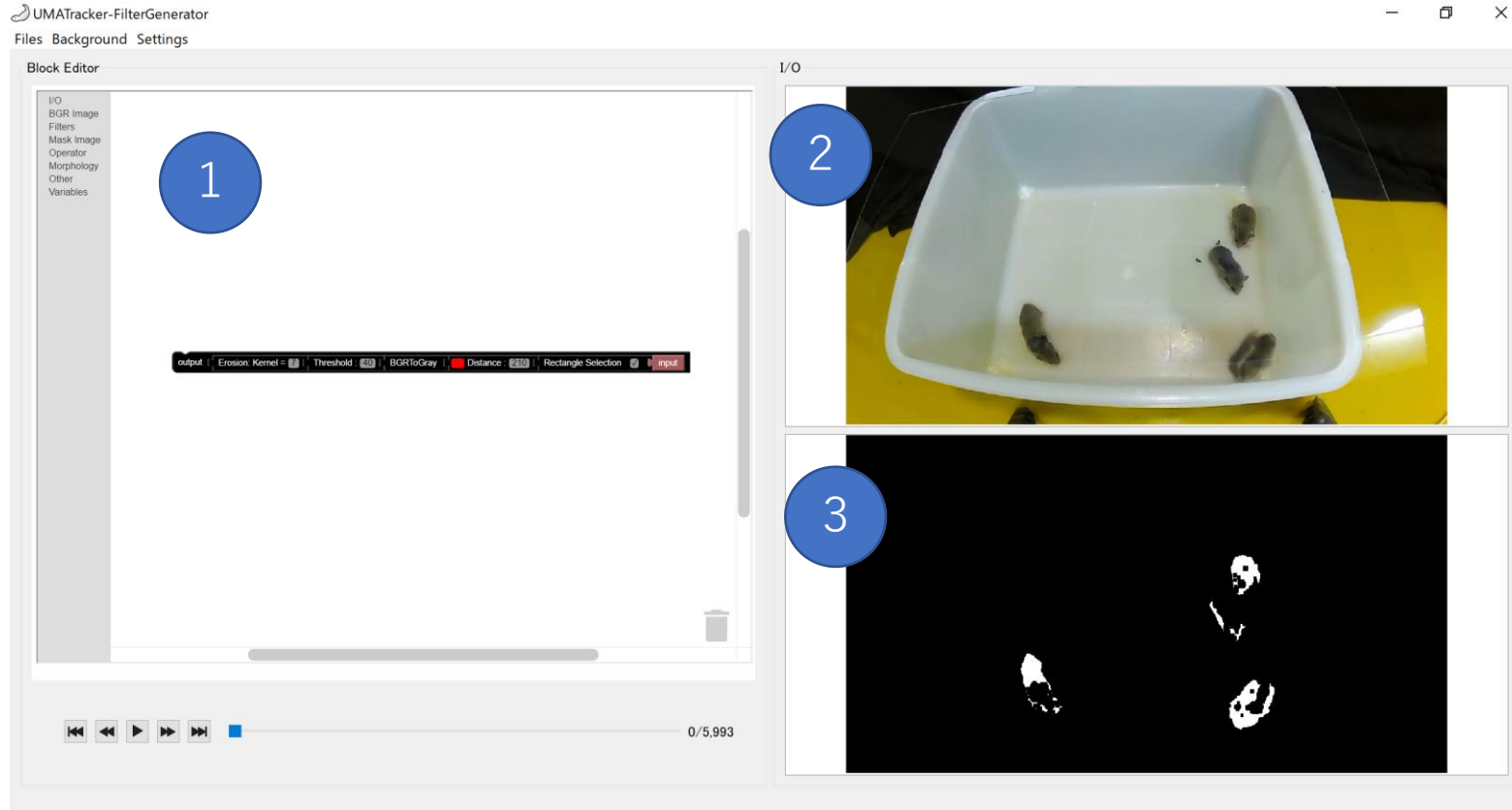
1. FilterGenerator

2. Tracking

3. Tracking Corrector

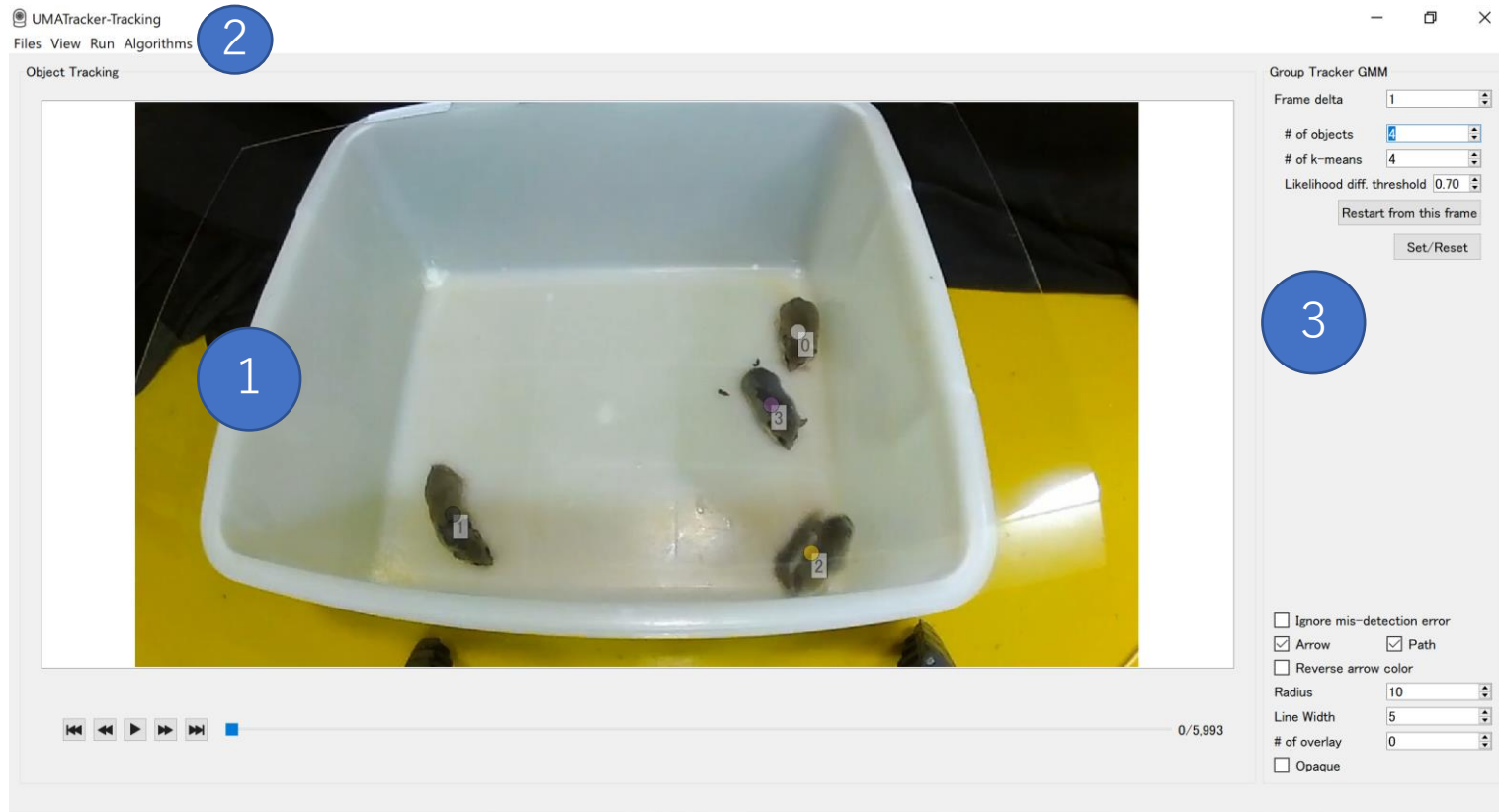
Filter Generator

追跡したい物体のみが白くなるようなフィルタを作成するために、パラメータを設定する。



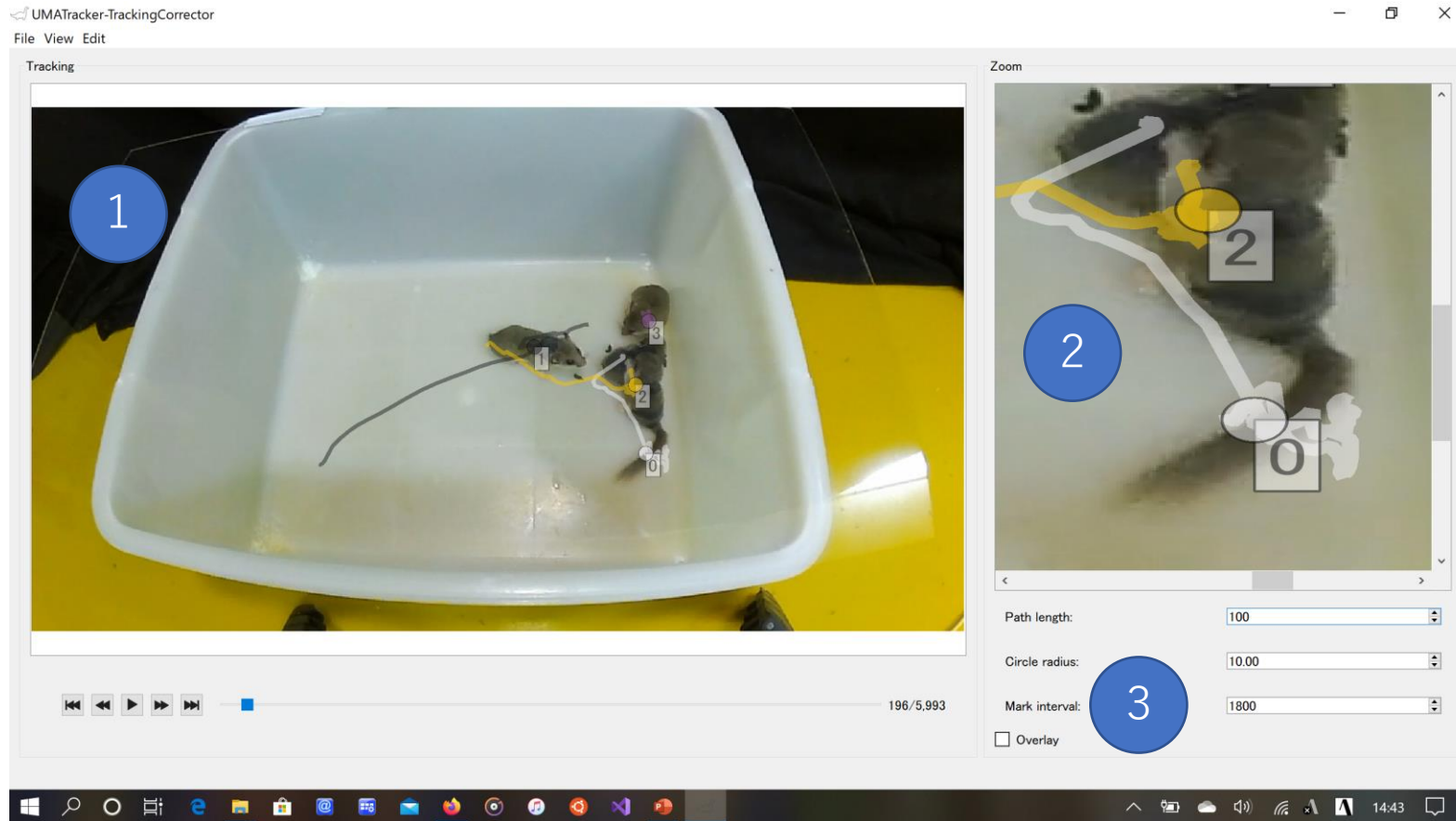
1. フィルタのパラメータ設定
2. 入力動画
3. フィルタの出力

Tracking



1. フィルタを適用した動画
2. アルゴリズム選択、追跡実行ボタン
3. ビジュアルに関するパラメータ設定

Tracking Corrector



1. トラッキング結果を適用した動画
2. 点の位置修正
3. ビジュアルに関するパラメータ設定

UMATrackerの問題点

複数体の同時追跡におけるUMATrackerの問題点

対象同士が隣接すると検出に不具合が起きる



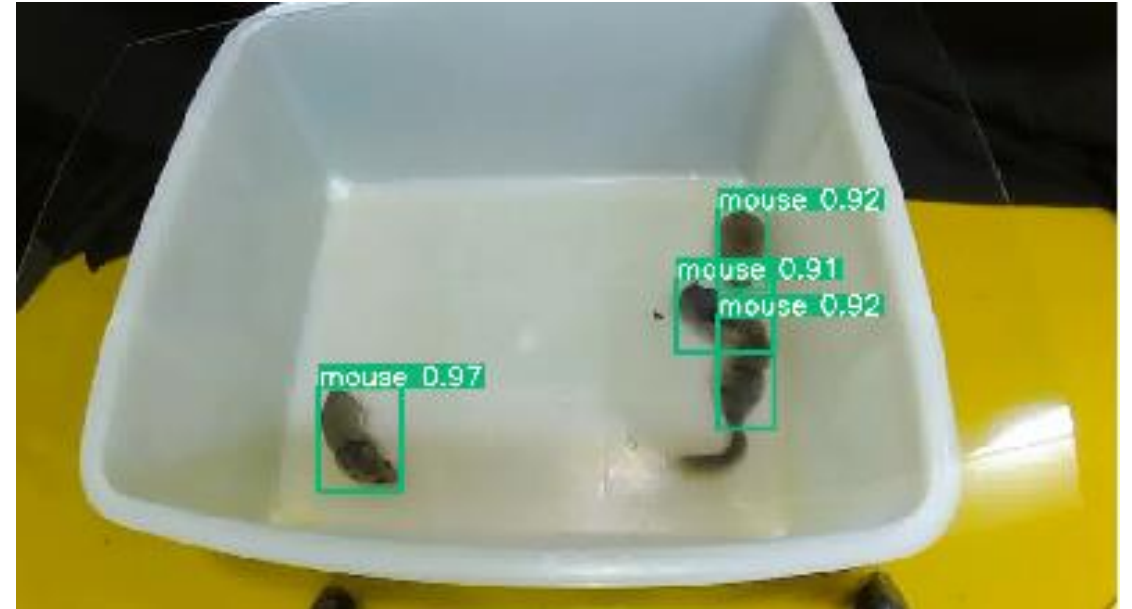
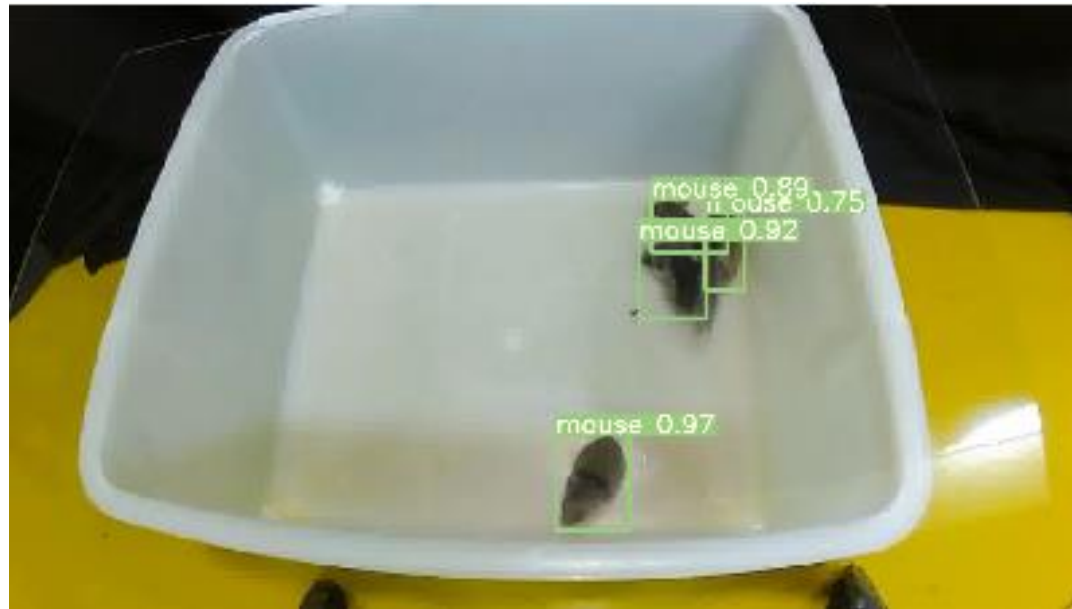
追跡失敗



深層学習による検出器の導入

深層学習による検出手法であるYOLOをUMATrackerのFilterGenerator工程に導入する

対象同士が隣接した場合でも区別して検出することができる



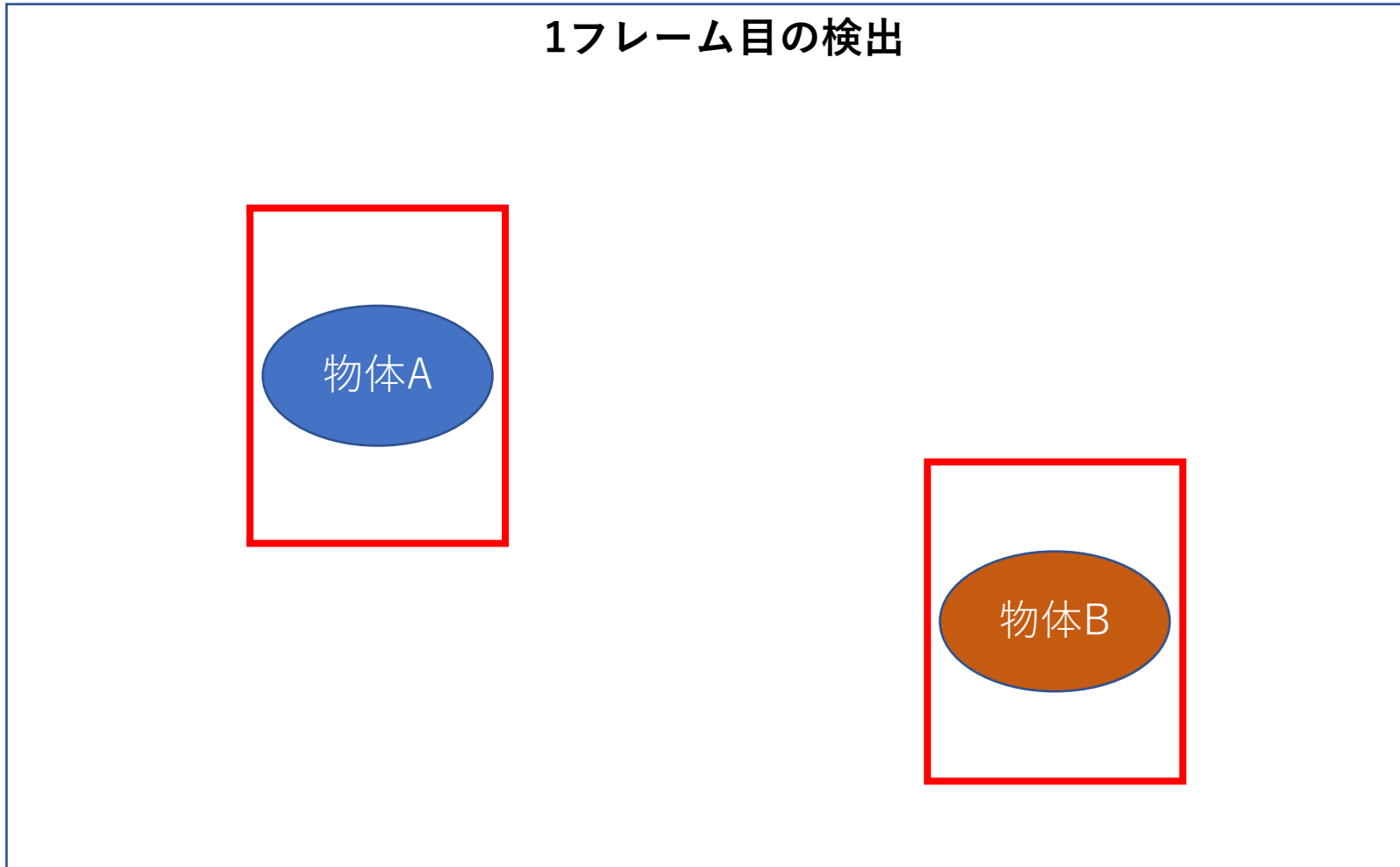
YOLOを用いた小動物追跡

YOLOを用いた追跡

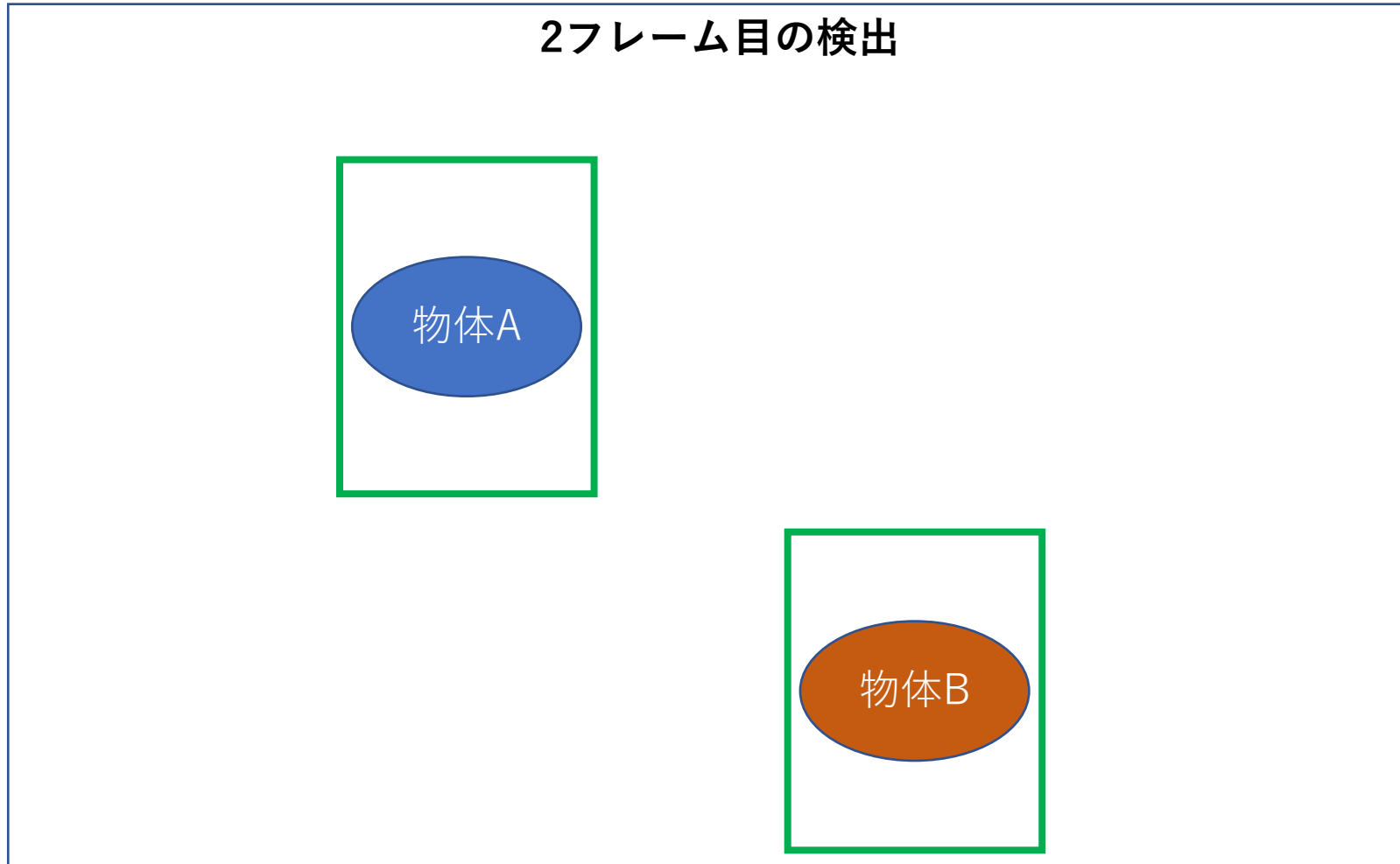
1. UMATrackerのFilterGenerator工程でYOLOによる検出
2. UMATrackerのTracking工程でハンガリーアルゴリズム[※]による
個体毎の検出ボックスの割り当てを行い追跡する

※ https://qiita.com/m__k/items/8e2cb9067ec5d720c30d

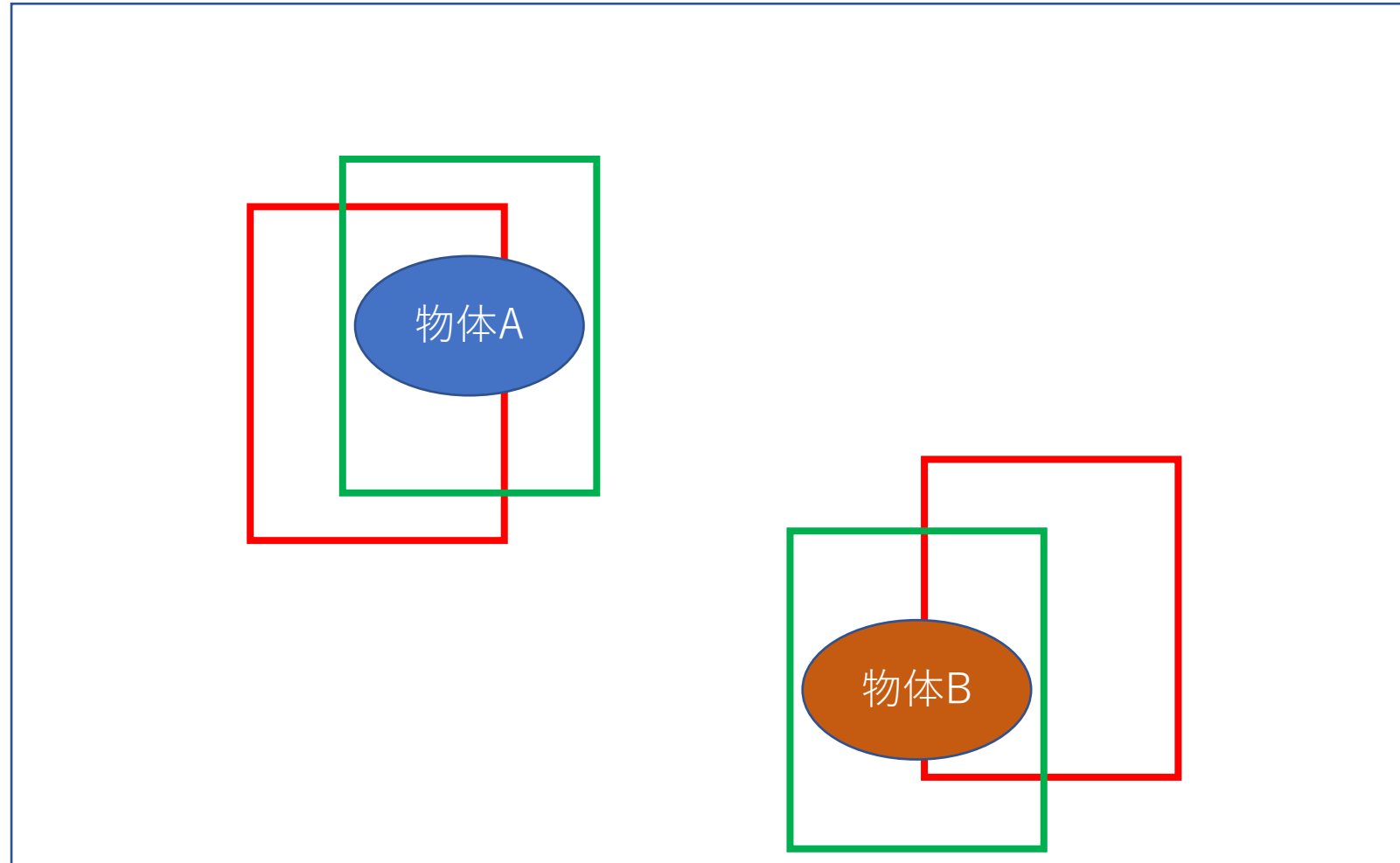
検出ボックスの割り当て



検出ボックスの割り当て



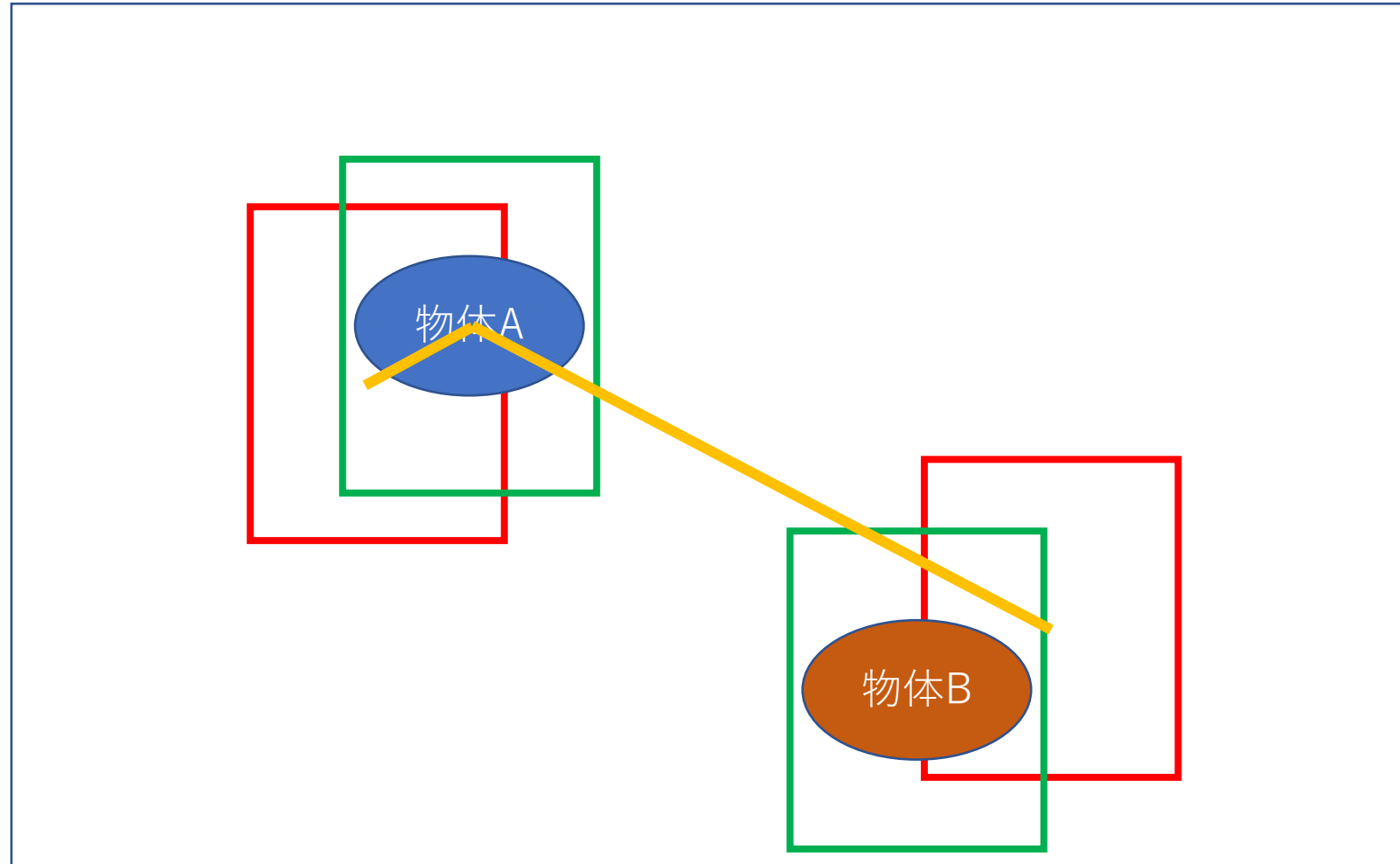
検出ボックスの割り当て



1フレーム目の
検出ボックス

2フレーム目の
検出ボックス

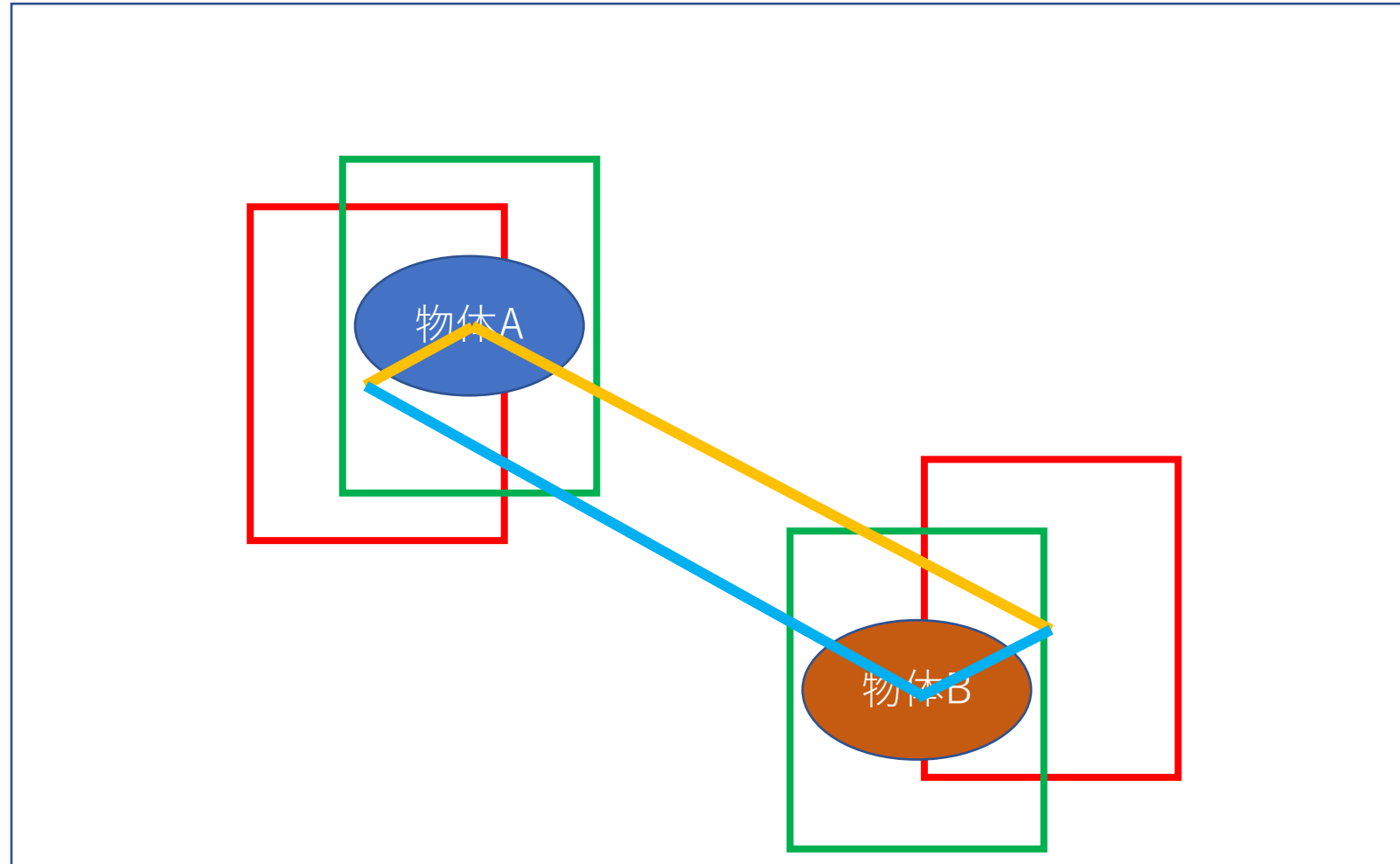
検出ボックスの割り当て



1フレーム目の
検出ボックス

2フレーム目の
検出ボックス

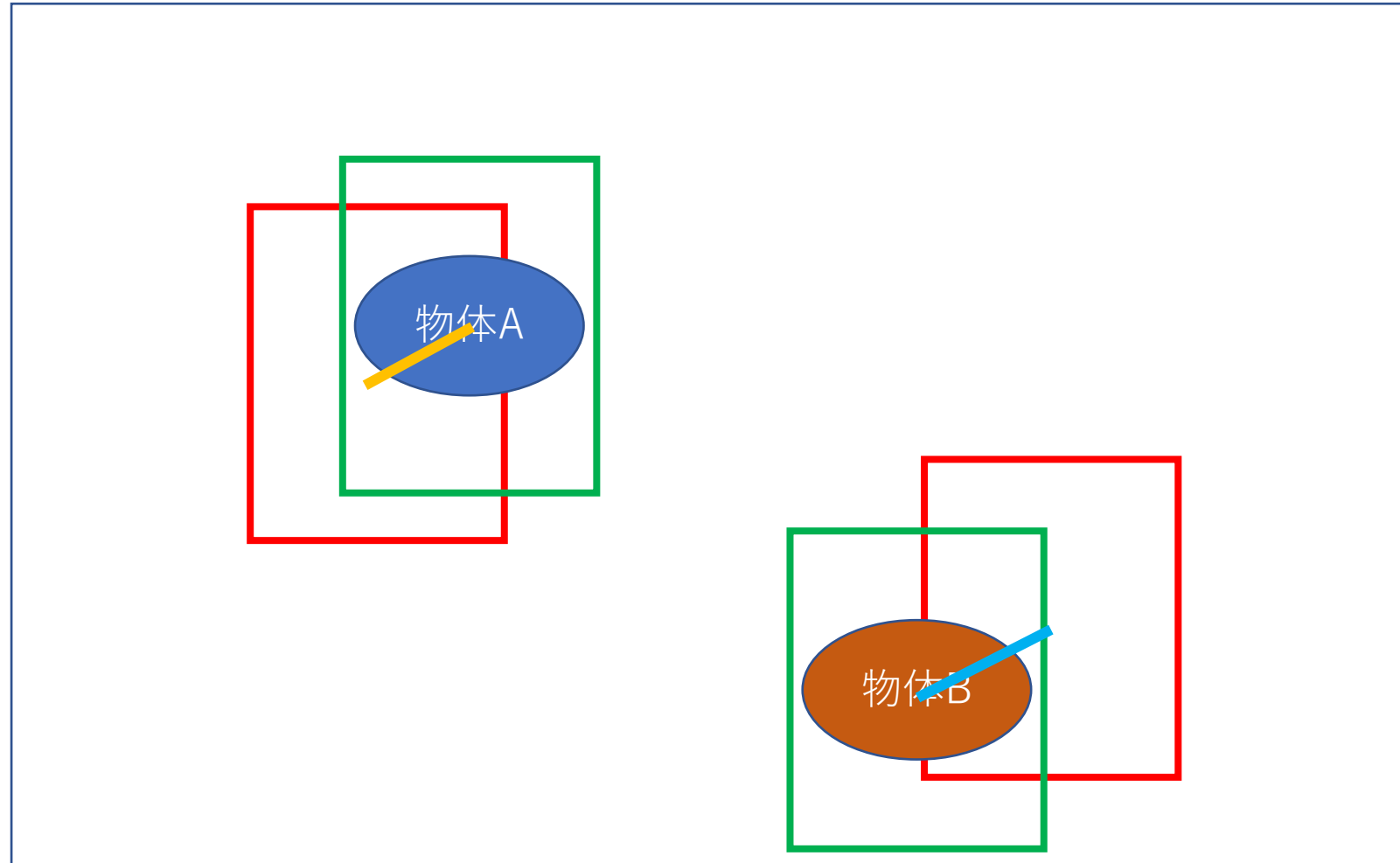
検出ボックスの割り当て



1フレーム目の
検出ボックス

2フレーム目の
検出ボックス

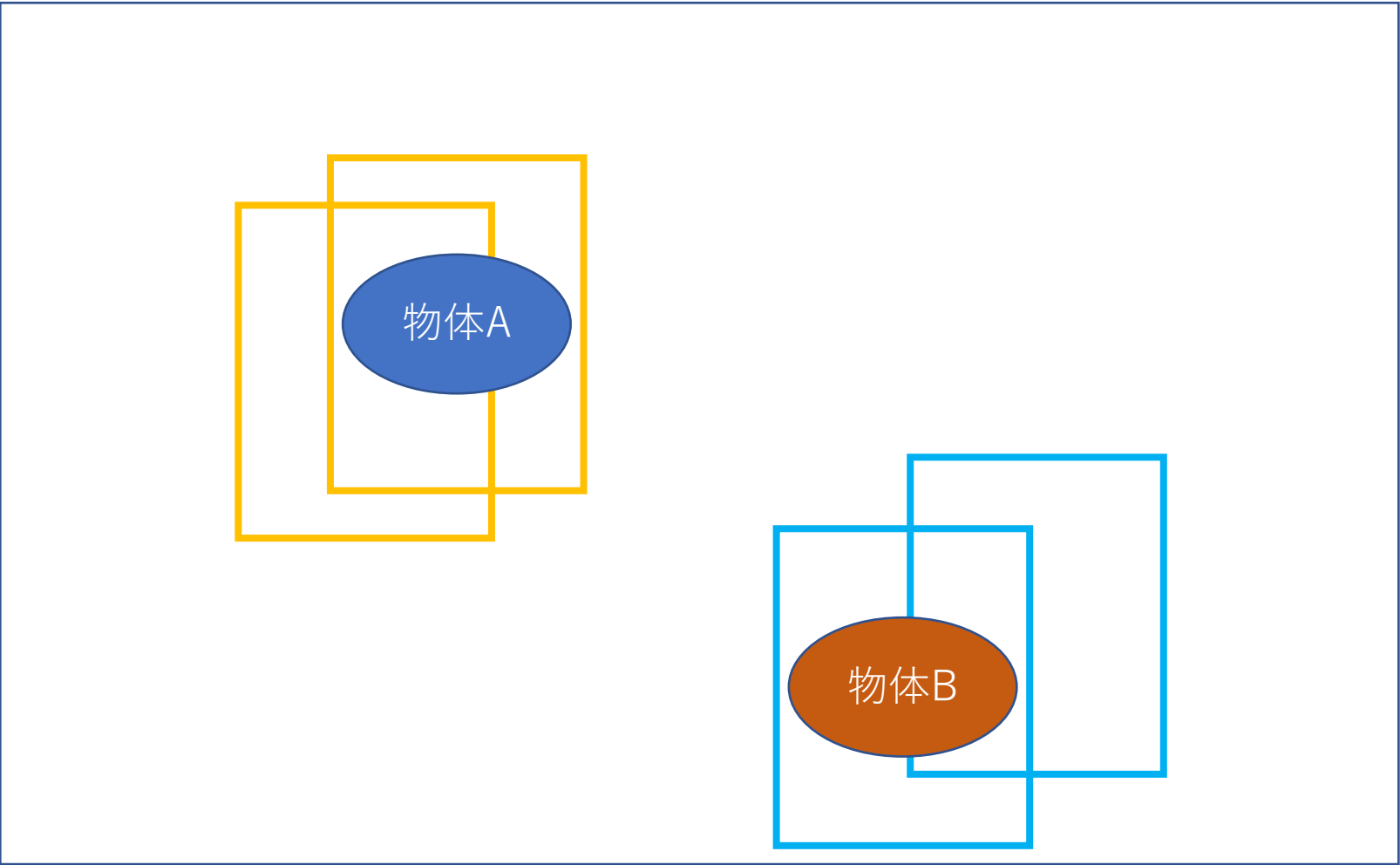
検出ボックスの割り当て



1フレーム目の
検出ボックス

2フレーム目の
検出ボックス

検出ボックスの割り当て



実験

実験

対象を見失う状況がない(オクルージョンがない)動画を用いて、4体のマウスを追跡する。

実装したプログラムと、UMATrackerの従来手法であるGroup Tracker GMM (以下GMM)と追跡精度を比較する。

追跡精度 = 手作業で入力した個体毎の中心座標と追跡結果との誤差

実験に用いたデータ

学習データ：

学習データ用動画から抽出した画像(データ拡張済み画像含む)1240枚

実験用動画：

4匹のマウスが映っている動画 300フレーム



学習データ用動画



実験用動画

実験結果

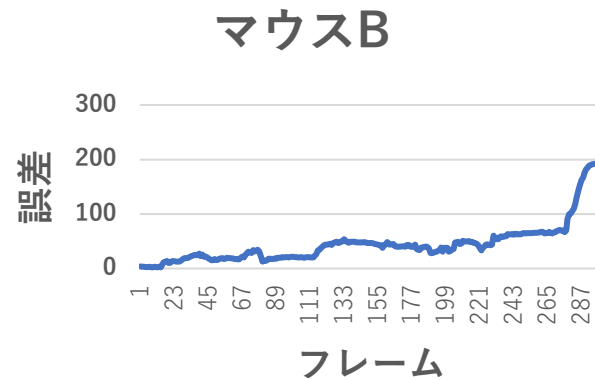
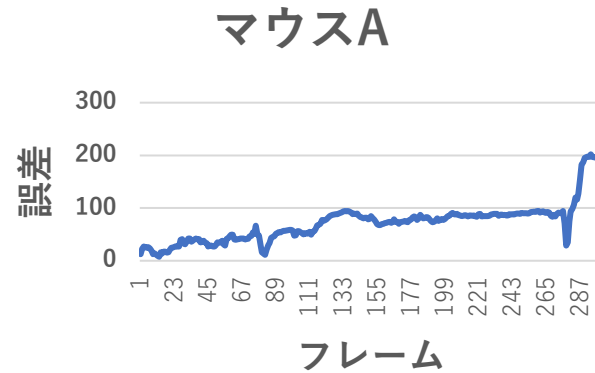


従来手法(GMM)

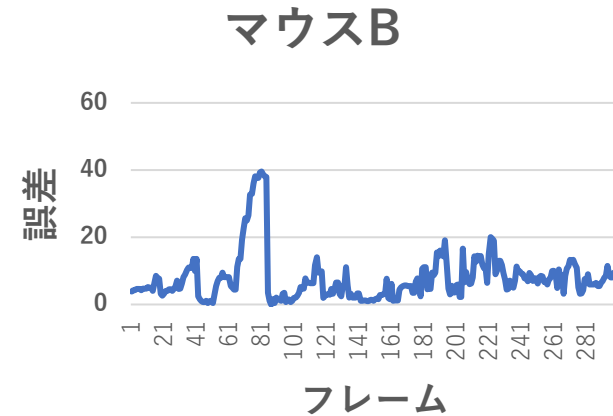
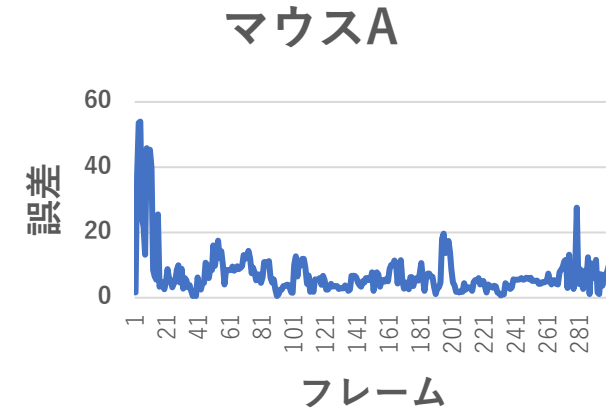


提案手法(YOLOを用いた追跡)

追跡誤差の推移比較



従来手法(GMM)



提案手法(YOLOを用いた追跡)

実験結果

表：追跡の平均誤差比較表

	従来手法(GMM)の平均誤差	YOLOを用いた追跡の平均誤差
マウスA	71.3	7.2
マウスB	45.5	7.9
マウスC	16.1	17.8
マウスD	7.0	9.8
全体の平均誤差	35.0	10.7

マウスAとマウスBが隣接

GMMによる追跡 → ×

YOLOを用いた追跡 → ○

結論

UMATrackerに深層学習による検出手法「YOLO」を導入



対象同士が隣接した時の追跡を改善した

今後の課題

- ・ 検出ボックスの割り当て方の改良
- ・ 学習データの工夫
- ・ UMATrackerのインターフェース改良

終