

令和3年度卒業論文

重なり滞留を起こす複数牛個体の高速追跡法

宮崎大学 工学部 情報システム工学科

AHMAD AMINNIN BIN MAT NOOR

指導教員 椋木 雅之 教授

# 概要

本研究では、重なり滞留が多く発生する牛の群れに対する複数物体追跡において、従来手法である SORT の高速な追跡処理を保ちつつ、SORT の追跡の誤り（ID スイッチ）数を削減するための改良を行う。複数物体追跡手法の一つである SORT は、比較的シンプルなアルゴリズムであり、高速な処理で追跡できるが、重なり滞留による ID スイッチをあまり考慮していなかった。そのため、複数牛個体の追跡のような現実の場面に SORT を適用すると ID スイッチが頻発するという問題点があった。ID スイッチの原因は、重なり滞留によって追跡対象が長時間検出されない点にある。ID スイッチを削減するために、本研究では、滞留する牛に対処するための「滞留」状態の導入、「滞留」状態での追跡データの保持と追跡 ID の結合、「滞留」状態での位置予測の停止、画像の拡大と再検出という機能を SORT に追加した。実験においては、提案手法と SORT、SORT の性能を向上させるために外観情報も使用する DeepSORT を牛の群れの動画に適用し、ID スイッチの発生回数により評価した。その結果、SORT で 48 回、DeepSORT で 12 回発生していた ID スイッチの回数を、2 回まで削減することができた。また処理速度についても、DeepSORT より速く、SORT と同程度に高速で、リアルタイム処理に必要な 30FPS を超えることを示した。このように、提案手法は SORT の高速性を維持しつつ、DeepSORT を超える追跡精度を達成した。今後の課題としては、さらなる検出精度の向上が挙げられる。実験では提案手法でもまだ 2 回の ID スイッチが起こったが、物体検出器の精度を上げることができれば、追跡精度もより向上できると考えられる。

# 目次

1 はじめに.....	1
2 複数物体追跡の従来研究.....	3
2.1 複数物体追跡.....	3
2.2 SORT.....	3
2.2.1 物体検出.....	4
2.2.2 追跡 ID の割り当て.....	4
2.2.3 追跡 ID の作成と削除.....	5
2.2.4 カルマンフィルタ予測.....	5
2.3 DeepSORT.....	6
2.4 従来手法の問題点.....	6
3 複数牛個体の高速追跡法.....	7
3.1 複数牛個体追跡.....	7
3.2 複数牛個体追跡に対する重なり滞留問題.....	8
3.3 重なり滞留への対処.....	9
3.3.1 画像の拡大と再検出.....	10
3.3.2 「滞留」状態の導入.....	12
3.3.3 追跡 ID の結合.....	12
3.3.4 カルマンフィルタ予測の停止.....	13
3.3.5 リフレッシュレート.....	14
4 実験評価.....	15
4.1 提案手法の評価.....	15
4.2 提案手法の各処理追加の効果.....	19
4.3 考察.....	21
5 おわりに.....	23
謝辞.....	24
参考文献.....	25

# 1 はじめに

複数物体追跡（Multiple Object Tracking ; MOT）は、映像に写っている複数の物体を追跡する手法の総称である。例えば、牛の群れの行動を分析するなど、複数物体追跡は多種多様な活用方法が考えられる。複数物体追跡手法の一つである SORT[1]は、深層学習で検出した追跡対象物体の位置をカルマンフィルタを用いて予測し、位置の近いもの同士をハンガリアン法により対応づけることで追跡を行う。比較的シンプルなアルゴリズムであり、高速な処理で追跡できる。しかし、追跡する途中で、それぞれの追跡対象物体に割り当てた追跡 ID が別の追跡 ID に入れ替わってしまうというエラー（ID スイッチ）が発生し、追跡精度が低下する。SORT では、追跡対象が他の物体の後ろに重なると、追跡対象の検出が難しくなる。また、他の物体に重なった状態で立ち止まる（滞留する）と、長時間検出されず見失ってしまう。従来の複数物体追跡では、通路を歩く人の追跡など、常に動き続ける物体を追跡対象とすることが多く、重なり滞留をあまり考慮していなかった。そのため、現実の場面に複数物体追跡を適用すると ID スイッチが頻発するという問題点があった。

本研究では、重なり滞留が多く発生する牛の群れに対する複数物体追跡において、SORT の ID スイッチ数を削減するための改良を行う。具体的には、ID スイッチを削減するために、画像の拡大と再検出、滞留する牛に対処するための「滞留」状態の導入、「滞留」状態での追跡データの保持と追跡 ID の結合、「滞留」状態での位置予測の停止という機能を SORT に追加する。ID スイッチの原因は、重なり滞留によって追跡対象が長時間検出されない点にある。画像の拡大と再検出によって、検出精度を上げる。この際、処理の高速性を保つため、一定間隔離れたフレームに対して処理を適用する。また、重なりが生じて牛が隠れて検出されなくなると、SORT では追跡を打ち切っていた。提案手法では「滞留」状態を導入し、追跡データを保持し続けることで、再度牛が検出された際に追跡を継続できるようにする。さらに、滞留している牛に対して、カルマンフィルタによる位

位置予測が外れることが多いため、滞留と判断された牛に対しては位置予測を停止する。この提案手法により、高速性を保ちつつ、SORTの追跡精度を向上させる。

以下、2章では複数物体追跡に対する従来研究の概要と問題点について述べる。3章では、本研究で対象とする複数牛個体追跡にある重なり滞留問題の対処法を提案する。4章では、提案手法の効果を調査するために行った実験について述べる。最後に、5章では、提案手法の結果を要約し、将来の改善について述べる。

## 2 複数物体追跡の従来研究

### 2.1 複数物体追跡

複数物体追跡とは、動画像中に含まれている複数の物体を同時に追跡する技術である。複数物体追跡には、バッチ型の処理とオンライン型の処理という二つの分類がある。バッチ型は、高精度を目指し、動画像の全てのフレームをまとめて追跡処理をする。一方、オンライン型は、高速性を目指し、現在のフレームと以前のフレームのみを使用し、追跡する。バッチ型では、現在と以前のフレームだけでなく、以降のフレームも追跡処理に必要なので、実時間での追跡が行えない。一方、オンライン型では、追跡精度が低いという問題があった。

これに対して、近年、オンライン型の一環である「検出による追跡 (Tracking by Detection)」というアプローチが注目されている。検出による追跡では、動画像の各フレームに対して物体検出を行い、得られた物体領域を時間方向に対応づけることで追跡を行う。物体検出技術の発展により、物体検出が高精度で行えるようになり、時間方向への比較的単純な対応づけ処理により、高速かつ高精度に追跡が行えるようになった。このアプローチで、SORT や DeepSORT といった手法が提案されている。

### 2.2 SORT

Simple Online and Real Time Tracking (SORT) [1]は検出による追跡のアプローチを取る一手法である。SORT の流れを図 1 に示す。各フレームに対して物体検出を適用する。物体検出結果は、物体を囲う矩形 (バウンディングボックス; bbox) で表現される。bbox に対して、それまでの追跡結果 (軌跡) の予測位置 (後述) と比較して、ハンガリアン法により最適になるよう追跡 ID を割り当てる。軌跡と対応づいた bbox は、その軌跡に追加して現フレームの情報を更新する。軌跡と対応しない bbox は新規 ID を割り当てて、新たな軌跡を作成する。bbox と対応しない軌跡は、一定期間の後、削除する。次フレームの

処理の前に、軌跡の最新の位置を元にカルマンフィルタで次フレームでの位置を予測する。  
以下に、各処理について述べる。

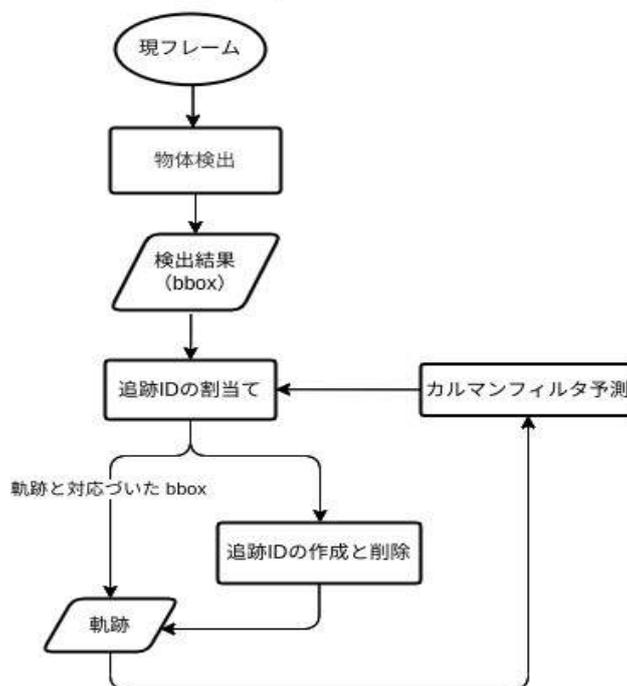


図 1: SORT の流れ

### 2.2.1 物体検出

最初のステップでは、物体検出手法を適用し、現フレームから追跡対象物体を検出する。例えば、追跡対象物体の bbox を検出するために、YOLO (You Only Look Once) [2] [3]が使用される。YOLO は、深層学習ネットワークを使用して、物体の位置の特定と検出した物体のクラス分類が行える高速な物体検出手法である。YOLO の入力画像であり、出力は bbox と bbox のクラス確率 (信頼度スコア; Confidence Score) である。0.5 より低い信頼度スコアを持っている bbox は無視される。

### 2.2.2 追跡 ID の割り当て

現フレームで検出した物体の bbox と、それまでに追跡した軌跡の予測した bbox (後述) の重なりに対して、ハンガリアン法により追跡 ID を割り当てる。ハンガリアン

法は、 $n$  人に  $n$  個の仕事を割り当てるとき、最も効率の良い割り当て方を求めるアルゴリズムである。各仕事に人は 1 人しか割り当てることができない状態で、全ての仕事に人を割り当てるとき、一番低コストの組み合わせを求める [4]。bbox の重なり部分は IoU スコアで表す。図 2 に IoU の式を示す。IoU (Intersection of Union) は、2 つの領域がどのくらい重なっているかを表す指標で、2 つの領域の共通部分を和集合で割ったものである。

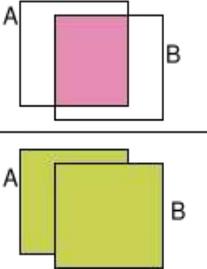
$$\text{IoU} = \frac{\text{AとBの共通部分}}{\text{領域A} + \text{領域B} - \text{AとBの共通部分}}$$


図 2: IoU の式

現フレームの bbox と予測結果が一致している場合は、IoU の最大値 1 になる一方、両方が重ならない場合、ゼロになる。SORT では、IoU スコアが閾値 0.3 以上のみ有効として、軌跡に加える。

### 2.2.3 追跡 ID の作成と削除

物体がフレーム中に新たに入ってきた時は、新規 ID を作成する。新規 ID は、追跡対象物体が 3 フレーム連続で追跡できた場合のみ軌跡として追加される。これによって、物体検出器からの誤検出 (False Positive) を軌跡として誤って登録することを防止する。

逆に、追跡対象物体が一定期間  $T_{lost}$  の間検出されなかった場合、追跡対象がフレームから出たとみなして、その ID は削除される。

### 2.2.4 カルマンフィルタ予測

SORT では、カルマンフィルタを使用し、軌跡における最新の bbox の次フレームでの位置とサイズを予測する。カルマンフィルタは、直前までの情報と、今取得したデータを元に、もっとも適切なシステムの状態を推定する手法である。SORT では、等速度を仮定し、bbox の次フレームでの位置と大きさを予測する。また、現フレームでの検出結果を

利用して、速度パラメータを更新する。これを繰り返すことで、精度を上げながら位置を予測する。

## 2.3 DeepSORT

Simple Online and Real Time Tracking with a Deep Association Metric (DeepSORT) [5]は外観情報も使用して、SORT の性能を向上させた手法である。外観特徴は追跡 ID を割り当てる時に使う。DeepSORT は追跡対象の外観特徴を CNN を使用して抽出する。追跡 ID を割り当てる時は、この外観特徴ベクトルが類似している bbox を優先的に割り当て、一致しない bbox のみ IoU スコアでマッチングさせる。外観特徴の採用により、より長いオクルージョン（他の物体と重なり合う）期間を通じて物体の追跡ができ、ID スイッチ数も効果的に減らすことができる。論文[5]の実験結果によると、DeepSORT は SORT の ID スイッチ数を 45%削減できた。

## 2.4 従来手法の問題点

SORT は単純なアプローチにより、高速な処理で良好なパフォーマンスが得られるが、オクルージョンが起こると、ID スイッチがしばしば発生し、追跡の精度が下がる。また、YOLO が追跡対象物体の bbox を検出できない場合、SORT はこの追跡対象物体を追跡できない。この状況では、もし次フレームでこの追跡対象物体が YOLO から再検出されると、追跡対象物体は新たな ID を取得する。この種の ID スイッチに関しても、SORT での追跡精度の低下につながる。

SORT [1]では、短期および長期のオクルージョンの問題は無視するとしている。これらの問題はあまり起こらない上、解決しようとしても、追跡フレームワークに望ましくない複雑さが生じるためであるとしている。しかし、本研究で対象とする重なり滞留のある牛の群れ動画では、オクルージョンが頻繁し、追跡精度への影響を無視できない。

DeepSORT のアプローチは、ID スイッチ数を減らすためには良いが、処理時間や必要なメモリ量が多く、リアルタイム処理に不向きとなる。

## 3 複数牛個体の高速追跡法

### 3.1 複数牛個体追跡

肉牛や乳牛の酪農では、農業資源である牛の管理が必要である。したがって、牛の行動を自動で監視することがますます重要になっている。食糧需要が増えるにつれ、農場の規模も平均的には大きくなる。この結果として、農家は牛の群れを観察する時間が取れなくなり、情報技術への依存度が高まっている[6]。

複数牛個体を自動で監視するには物体追跡手法が使用される。例えば、Stephensonら[7]は、GPS（Global Positioning System；全地球測位システム）を使った追跡を行っている。GPSを個々の牛の首輪につけて追跡することで、牛の行動を記録できる。しかし、多数の牛それぞれにGPSを装着させるのは、コストと設置の手間がかかる。一方、Daoら[6]は農場内に固定設置した複数のカメラにより、複数牛個体を追跡している。複数カメラの場合、カメラを適切な場所に設置し、すべてのカメラを同期する必要がある。この場合、処理が複雑となり計算量もかかるため、実時間での処理が難しくなる。

単一カメラで複数牛個体を追跡する場合、処理時間の問題の回避はできるが、重なり滞留が大きな問題となる。牛は社会的な動物であり、放牧地で放牧する場合、通常はグループ内で行動する[7]。そのため、群れ内で牛が休息や移動をバラバラにすることによって、重なり滞留がよく発生する。単一カメラでの複数牛個体追跡では、カメラの前で写っている牛だけが検出でき、後ろに重なった牛は検出されなくなる。そのため追跡が困難になる。本研究は、単一固定カメラでの複数牛個体追跡において、重なり滞留問題を低減する方法を提案する。

## 3.2 複数牛個体追跡に対する重なり滞留問題

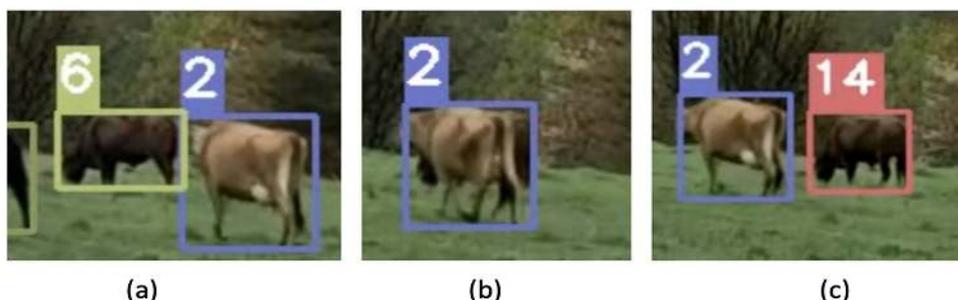


図 3: SORT の重なり滞留の追跡

図 3 に、重なり滞留が生じた状況での SORT の追跡結果を示す。入力動画は[8]から取ったものである。図 3 (a) ように動く追跡 ID 2 の牛と滞留する追跡 ID 6 の牛がいる。重なっている際の図 3 (b) では、後ろに隠れている牛 6 が YOLO で検出できなくなる。この時、追跡 ID 6 の bbox の情報と速度がカルマンフィルタで更新できないため、一定の速度で予測が続けられる。検出できないフレームが長く続くと、予測位置が実際の物体の位置と外れてしまう。そうならないよう、SORT は追跡対象物体が  $T_{lost}$  フレームの間検出されない場合、その追跡対象物体の軌跡を終了させる。これによって、長期間にわたるカルマンフィルタ予測による無制限の位置ずれの増加が防止される。すべての SORT の実験で、 $T_{lost}$  は 2 つの理由で 1 に設定されている。第一に、実際の運動に対して、等速モデルは正確ではない。第二に、SORT は主にフレーム間の追跡に注目しているため、重なりによる物体の再識別は SORT の範囲を超えている。さらに、失われた追跡対象物体を早期に削除すると、処理効率が向上する。追跡対象物体が再登場した場合、SORT は新しい ID で暗黙的に追跡を再開する[1]。このため、図 3 (c) の牛 6 が新規 ID 14 に変わってしまう。

SORT は追跡処理の簡易性と高速性を重視しているため、重なり滞留への対する対処は考えられていない。このため、重なり滞留が SORT の課題になる。処理速度を落とさずに、元の SORT の重なり滞留による ID スイッチ数を減らすために、新しい対処法が必要である。

### 3.3 重なり滞留への対処

本研究では、重なり滞留が多く発生する牛の群れに対する複数物体追跡において、SORT の ID スイッチ数を削減するための対処法を提案する。提案手法の流れを図 4 に示す。太文字は SORT に追加された機能である。

物体検出においては、画像の拡大と再検出を追加する。本研究で物体検出に用いる YOLO では、手前に大きく牛が撮影されている場合、一部が隠れていても検出可能であるが、遠くに小さく撮影されている牛では、検出に失敗することが多い。画像を拡大して再検出することで、検出失敗を低減する。

追跡 ID の割り当て時には、追跡中の軌跡の中で bbox と対応づかなかったものについて、SORT では削除候補としていたが、提案手法では「滞留」という新たな状態を割り当て、軌跡の情報を保持し続けるようにする。これにより、長時間滞留している牛についても、継続して追跡が行えるようにする。

追跡 ID の作成と削除の処理では、新たに追跡 ID の結合処理を追加する。重なりにより牛の一部が隠されると、一頭の牛の見えている部分に対して別々に複数の bbox として検出されてしまうことがある。一旦このような bbox が検出され別々の軌跡として追跡されると、重なりが解消した後も、複数の軌跡として追跡が続き、ID スイッチの原因となる。これを避けるために、「滞留」状態の軌跡に対し、近くに新規 ID の bbox が検出された場合、追跡 ID を結合して一つにする処理を追加する。

カルマンフィルタ予測では、予測の停止を追加する。「滞留」状態の軌跡に対して、等速運動で長期間予測を行うと、滞留している牛の位置から予測位置が大きく外れてしまう。これを避けるために、「滞留」状態の軌跡に対してはカルマンフィルタ予測を停止する。

画像の拡大と再検出、および追跡 ID の結合は、一定間隔離れた時間（リフレッシュレート）毎に行う。これらの処理は、やや処理時間がかかるため、毎フレーム処理すると処理速度が大きく低下する。精度と処理速度のトレードオフを考慮して、適切なリフレッシュレートで処理を適用する。

以下に、各処理について述べる。

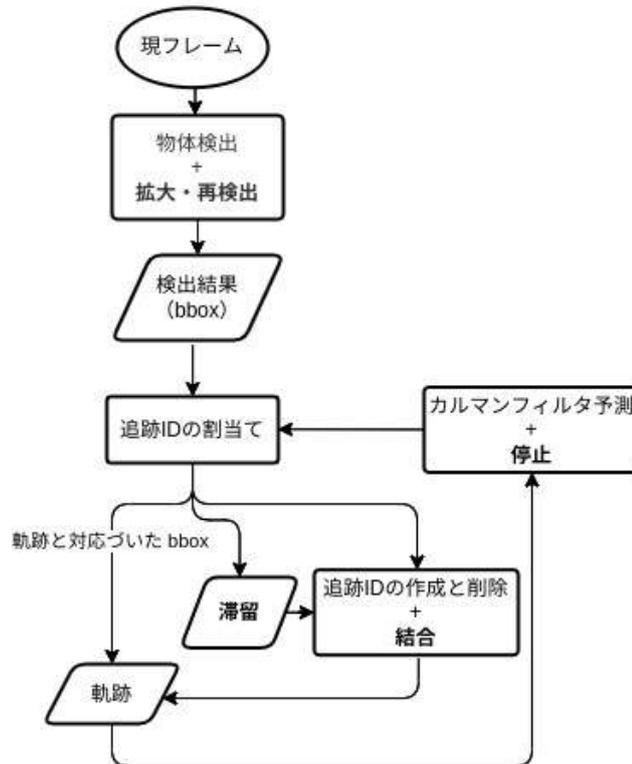


図 4: 提案手法の流れ

### 3.3.1 画像の拡大と再検出

画像拡大機能は[9][10]からヒントを得た。カメラから遠く離れて小さく撮影されている牛では、YOLOで物体検出する際、信頼度が低くなる。信頼度が閾値0.5より小さければ、システムには無視される。この検出精度の低下により本来同じ追跡対象である牛が、新規IDを持つ別の牛として扱われ、IDスイッチが起こる。

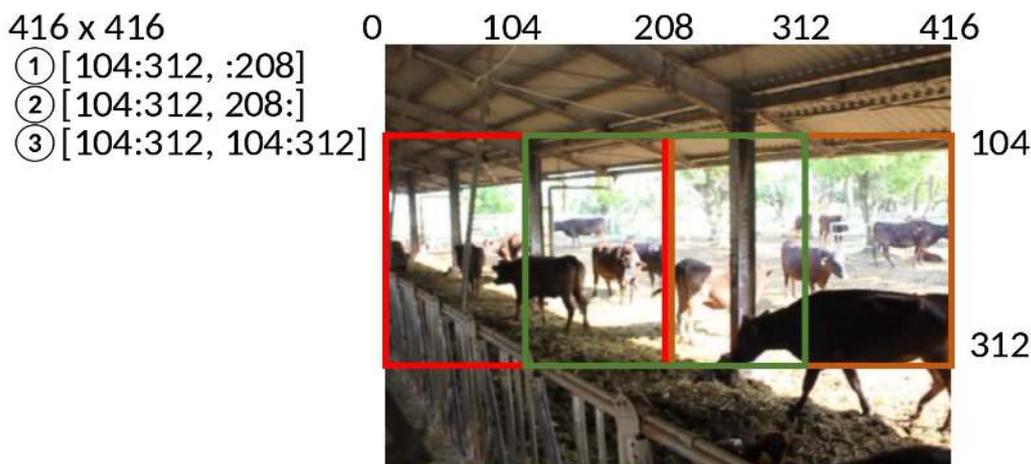


図 5: 拡大する 3 つの部分

YOLO への入力画像の大きさは一定(416x416 画素)であるので、画像の一部を切り取り、拡大して YOLO への入力とする。具体的には、フレーム 1 枚に対して、図 5 のように、3 つの部分 (左、真ん中、右) を切り取る。切り取り画像の大きさは、元画像の縦横 1/2 である。それぞれの切り取り画像を縦横 2 倍に拡大して YOLO に与え、再検出を行う。再検出結果の bbox 座標を 2 で割って、画像の元の位置に挿入する。これにより、再検出により検出できた bbox を元画像に追加することができる。一方、元画像でも再検出でも bbox が検出された場合、同じ牛を二重に検出してしまう。これに対処するために、非極大値の抑制 (英: Non-Maximum Suppression ; NMS) [11]を使用する。非極大値の抑制は、重なり合う bbox の中から最も高い信頼度スコアを持つ bbox のみを選択することを目的としている。非極大値の抑制の処理例を図 6 に示す。低い信頼度スコアを持っている bbox が削除され、高い信頼度スコアまたは適切な bbox を持っている bbox のみが残る。



図 6: 非極大値抑制

画像を拡大することで、高い信頼度が得られ、正確な bbox の座標が得られる。しかし、画像拡大の処理は元画像に対する処理に加えて3回の検出処理を追加することになり、計算量が多く処理速度が落ちるという問題が生じる。これについては、後述のリフレッシュレートの導入で対処する。

### 3.3.2 「滞留」状態の導入

SORT では、追跡 ID の割り当て時に、検出した bbox を割り当てられなかった軌跡は、削除候補となり、 $T_{lost}=1$  フレーム後には削除していた。この場合、本研究が対象としている重なり滞留が多く発生する場面では、隠れ等により検出に失敗する度に新規 ID を割り当ててしまい、ID スイッチが頻発する。これを避けるために、bbox を割り当てられなかった軌跡に対して「滞留」という状態を割り当てる。

「滞留」状態の軌跡については、削除はせず、軌跡中の最新の bbox の情報を保持し続ける。追跡 ID の割り当て時には、「滞留」状態の軌跡も割り当ての候補として扱う。これにより、滞留していた牛が再度検出された際には、その近くにある「滞留」状態の軌跡が割り当てられ、正しく追跡が続けられる。また、「滞留」状態の軌跡については、後述の処理で滞留した牛がその場所に居続けることを考慮して扱う。

軌跡に bbox が割り当てられないケースは、隠れ等で物体検出が失敗した場合以外に、追跡対象がカメラの視野外に出た場合もある。本研究では、追跡対象である牛が視野に入ったりするのは、フレームの端のみと仮定し、軌跡の最新の bbox 位置がフレームの周辺部分であった場合は、「滞留」状態とはせず、その軌跡の追跡を終了（ID を削除）する。

### 3.3.3 追跡 ID の結合

重なりにより牛の一部が隠されていると、見えている部分に対して別々に複数の物体として検出されることがある。図7は、一頭の牛が木の後ろに移動したフレームでの検出結果である。頭の部分に追跡 ID 9 の bbox が検出され、新たな軌跡が作成されている。一方、追跡 ID 5 の軌跡も残されている。

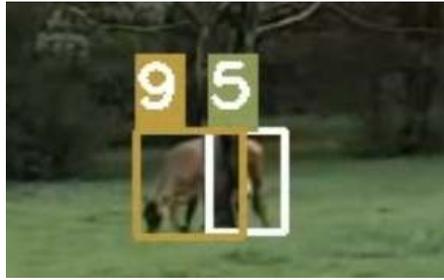


図 7: 一頭の牛に対する複数の軌跡

一旦このような bbox が検出され別々の軌跡として追跡されると、重なりが解消した後も、複数の軌跡として追跡が続く。そのため、一つの bbox を複数の軌跡が取り合う状況が生じ、ID スイッチが頻発する。この時、bbox に割り当てられなかった方の軌跡は「滞留」状態となる。これを避けるため、「滞留」状態の軌跡に対し、近くの新規 ID の bbox との距離を計算する。距離が閾値以下であった場合、追跡 ID を結合して一つにする。

### 3.3.4 カルマンフィルタ予測の停止

「滞留」状態の軌跡に対して、等速運動で長期間予測を行うと、滞留している牛の位置から予測位置が大きく外れてしまう。これを避けるために、「滞留」状態の軌跡に対してはカルマンフィルタ予測を停止する。

SORT では、等速運動を仮定してカルマンフィルタにより次フレームでの軌跡の bbox 位置を予測する。等速運動は、それ以前の bbox の追跡結果により推定される。そのため、追跡対象が滞留・静止していても、bbox の多少の検出結果の変動により、速度は 0 とならない。「滞留」状態で検出されていない牛については、等速運動の推定が更新されないため、検出されない間、等速で予測位置が移動することになる。これにより、長時間滞留している牛の軌跡において、カルマンフィルタによる予測位置が、滞留している牛の位置から大きく外れてしまう。図 8 は、カルマンフィルタ予測によって、追跡 ID 8 の牛の bbox の位置が実際の位置より外れる様子を示している。白い bbox は、その時 YOLO で検出されなく、カルマンフィルタにより予測された bbox を表す。図 8 のように、長時間重なり滞留が続くと、予測 bbox が滞留している牛から外れていく。

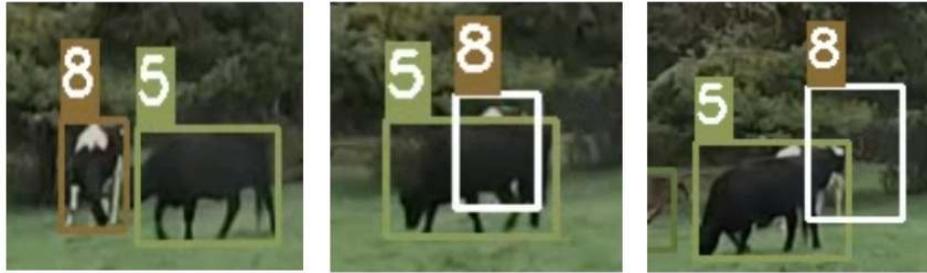


図 8: カルマンフィルタ予測による位置外れ

これを解決するために、「滞留」状態の軌跡については、カルマンフィルタの予測を停止し、以前の bbox の位置情報を保持し続ける。これにより「滞留」状態の軌跡では、bbox の予測位置は最後に検出された位置に固定される。滞留状態だった牛が動き出し、元の軌跡の位置の近くで再度検出されれば、同じ ID で追跡を継続できる。

### 3.3.5 リフレッシュレート

高速な処理を確保するために、3.3.1 節の画像の拡大と再検出、3.3.3 節の追跡 ID の結合は、一定間隔（リフレッシュレート）離れたフレームに対して処理を適用する。この際、追跡の精度と処理時間のトレードオフを考慮して、リフレッシュレートを定める必要がある。提案手法では、リフレッシュレートを 60 フレームとした。

## 4 実験評価

### 4.1 提案手法の評価

提案手法の効果を調査するために、牛の群れ動画に対して実験を実施した。動画の1フレームを図9に示す。動画は宮崎大学農学部住吉フィールドにおいてデジタルカメラで録画したものである。3分程度の長さの動画中から29秒（60 FPS、1772フレーム）を切り取り、処理対象とした。この動画に対して提案手法を適用し、動画内の複数牛を追跡した。比較のために、SORT 及び DeepSORT でも同様に処理した。

実験では、ID スイッチの回数と処理速度を評価する。ID スイッチは、追跡中に追跡対象物体の追跡 ID が変わることであり、追跡処理としては間違った結果である。本実験では、一時的な ID スイッチは無視し、60 フレーム以上同じ ID で追跡した軌跡について、目視で ID スイッチの回数を数えた。処理速度は1秒あたりの処理フレーム数（Frame per Second ; FPS）で表す。プログラム中で、処理時間と処理フレーム数から計算した。使用した計算機環境を以下に示す。YOLO による牛検出の際には、GPU 処理を用いている。

CPU	: AMD® Ryzen 9 3900x 12-core processor × 24
メモリ	: 128 GBytes
GPU	: NVIDIA GeForce RTX 2080 SUPER (メモリ : 8 GBytes)



図 9: 実験評価の入力動画

表 1: 提案手法の評価

	ID スイッチ数	FPS
SORT	48/15204 (0.32%)	1772 frames/48.51 seconds (36.52 FPS)
DeepSORT	12/15204 (0.08%)	1772 frames/82.49 seconds (21.48 FPS)
提案手法	2/15272 (0.01%)	1772 frames/49.17 seconds (36.04 FPS)

表 1 に実験結果を示す。ID スイッチ数の列は、ID スイッチ数を検出された物体の bbox 総数で割ったものである。例えば、SORT の場合は、15204 個の牛個体の bbox が検出されたのに対して、ID スイッチが 48 回が起こった。

表 1 より、提案手法は SORT と同じ程度(1.3%差) の FPS で、ID スイッチ数を 96.9% 減らすことができた。一方、DeepSORT は SORT の ID スイッチ数を 75%を減らすことが出来たが、処理速度は遅く、FPS が 41.1%低下した。

図 10、11、12 に SORT、DeepSORT と提案手法の重なり滞留が起こる時の追跡結果を示す。各図の(a)、(b)、(c) は重なり滞留が起こる前、起こっている状態、起こった後を示す。

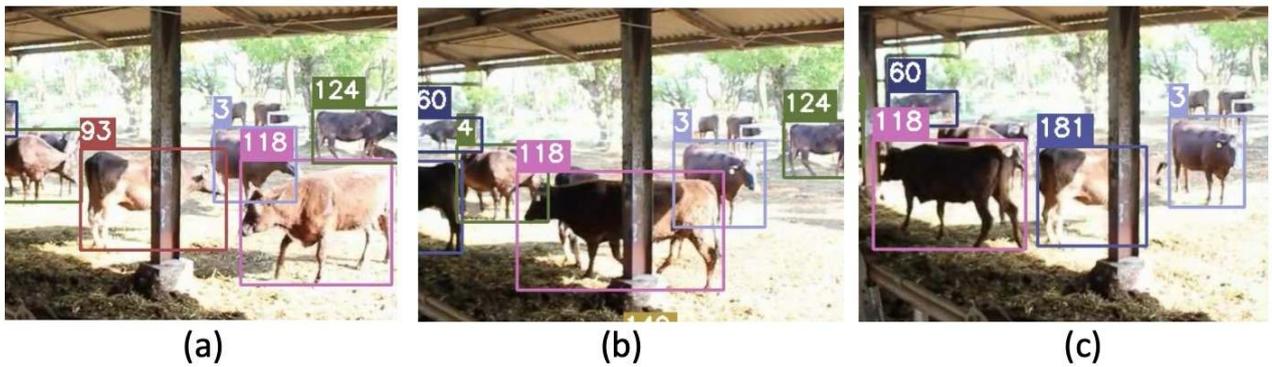


図 10: SORT の追跡結果

図 10 は SORT の追跡結果である。図 10(a) の滞留する追跡 ID 93 の牛は図 10(b) では完全に追跡 ID 118 の牛に隠れている。その後、図 10(c) の重なり滞留後では追跡 ID が 181 に変わっている。このように ID スイッチが起こっている。

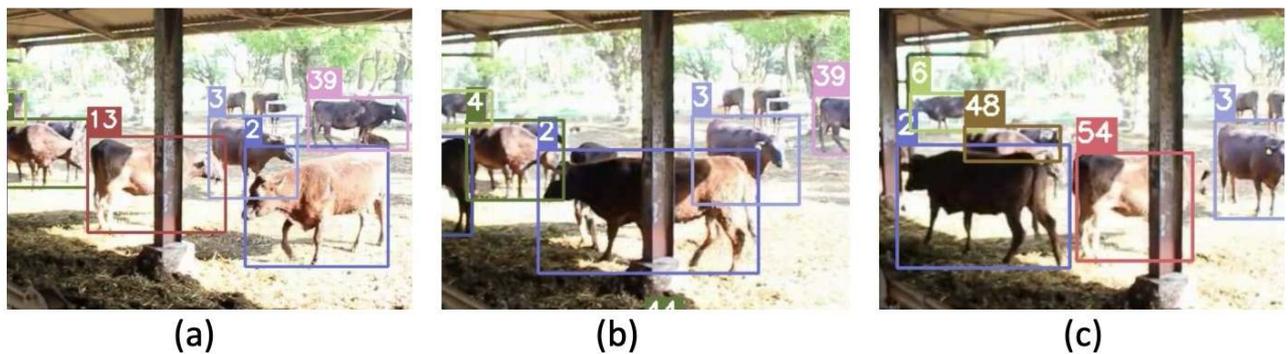


図 11: DeepSORT の追跡結果

図 11 は DeepSORT の追跡結果である。滞留する牛が追跡 ID 13 から新規 ID 54 に変わり、SORT の場合と同じように図 11(c) の時に ID スイッチが一回起こった。DeepSORT は外観特徴で追跡 ID をマッチングするが、この例では重なり滞留期間が長いため、DeepSORT の  $T_{lost}=60$  を越えた。そのため、図 11(c) の時には追跡 ID 13 がすでに削除されており、再度検出された牛が新規 ID となった。

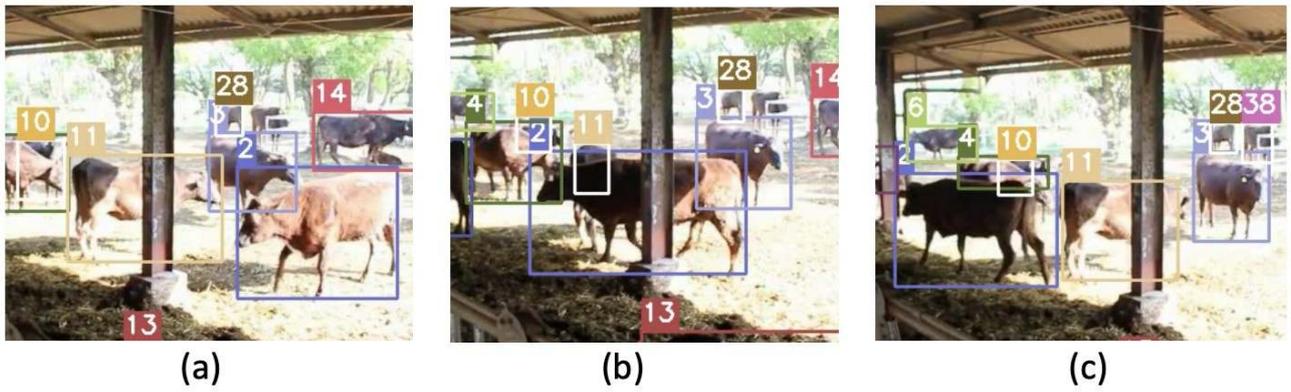


図 12: 提案手法の追跡結果

図 12 は 提案手法の追跡結果である。提案手法では、図 12(b) の時に重なった 追跡 ID 11 の bbox を保持し続ける。図 12(c) の時に再度検出した牛が新規 ID となるが、リフレッシュの時に追跡 ID の結合処理が行われ、元の追跡 ID 11 に戻った。これによって、重なり滞留による ID スイッチが防止できた。

## 4.2 提案手法の各処理追加の効果

提案手法により追加した各処理の効果を調査するために、実験を実施した。追加前の状態である SORT の処理に対して、提案したカルマンフィルタ予測の停止、追跡 ID の結合、画像の拡大と再検出の処理をこの順に追加し、ID スイッチ数の変化を調査した。結果を表 2 に示す。

表 2: 各処理追加の効果

	ID スイッチ数
SORT	48/15204 (0.32%)
カルマンフィルタ予測の停止の追加	6/15204 (0.04%)
追跡 ID の結合の追加	4/15204 (0.03%)
画像の拡大と再検出の追加 (提案手法)	2/15272 (0.01%)

表 2 より、SORT に対して、「滞留」状態を導入し、「滞留」状態の追跡軌跡についてはカルマンフィルタ予測を停止することで、ID スイッチの回数を 48 回から 6 回に、9 割削減できた。SORT では、オクルージョンによって、bbox が検出されず対応づけされないと、軌跡を削除する。その後、同じ牛が再度検出された場合、新規 ID に変わり、ID スイッチが生じる。提案手法では、「滞留」状態を導入し、bbox が検出されない場合も軌跡の情報を保持する。この時、カルマンフィルタ予測を停止することで、再度検出できた場合、元の追跡 ID に関連付け、重なり滞留が生じた牛の追跡を正しく続けることができる。

次に、追跡 ID の結合を追加することで、ID スイッチの回数を 6 回から 4 回に削減できた。追跡 ID の結合がないと、1 頭の牛を複数の bbox として検出し、その一部が「滞留」状態の軌跡として残り続ける場合がある。この「滞留」状態の軌跡の bbox 位置に、他の牛が移動してくると、ID スイッチが生じる。追跡 ID の結合を追加することで、このようなゾンビ軌跡を統合し、ID スイッチが生じる可能性を減らすことができる。

最後に、画像の拡大と再検出を追加することで、ID スイッチの回数を 4 回から 2 回に削減できた。画像の拡大と再検出により、検出された bbox 数が 68 増えた。これにより、検出もれが減り、ID スイッチを減らすことができた。

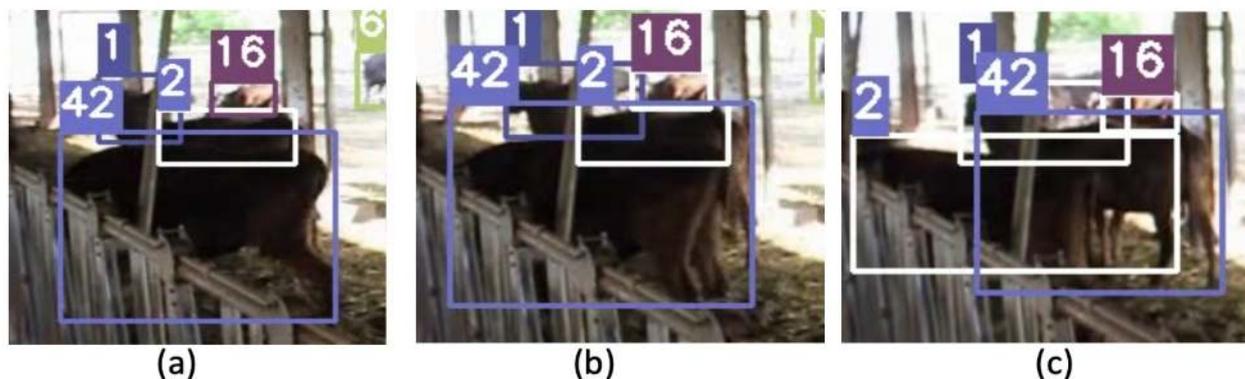


図 13: 提案手法に起こる ID スイッチ

提案手法では、ID スイッチの回数を 48 回から 2 回に削減できた。図 13 に提案手法で生じた ID スイッチを示す。ID スイッチが起こる原因は検出器の低精度または誤検出によると考えられる。図 13(a)の追跡 ID 42 と ID 2 の牛に着目する。図 13 (b) の時、ID 42 の bbox が非極大値抑制と IoU 計算によって、これら 2 頭の牛を 1 つの bbox として検出してしまふ。図 13 (c) の時は ID 42 の bbox が後ろの牛（旧 ID 2 の牛）に移動する。再検出された旧 ID 42 の牛が新規 ID となるが、リフレッシュ時の追跡 ID の結合処理により、「滞留」状態の ID 2 を割当ててしまふ。このような状況によって、提案手法では 2 回の ID スイッチ（ID 2 → ID 42、ID 42 → ID 2）が生じた。

### 4.3 考察

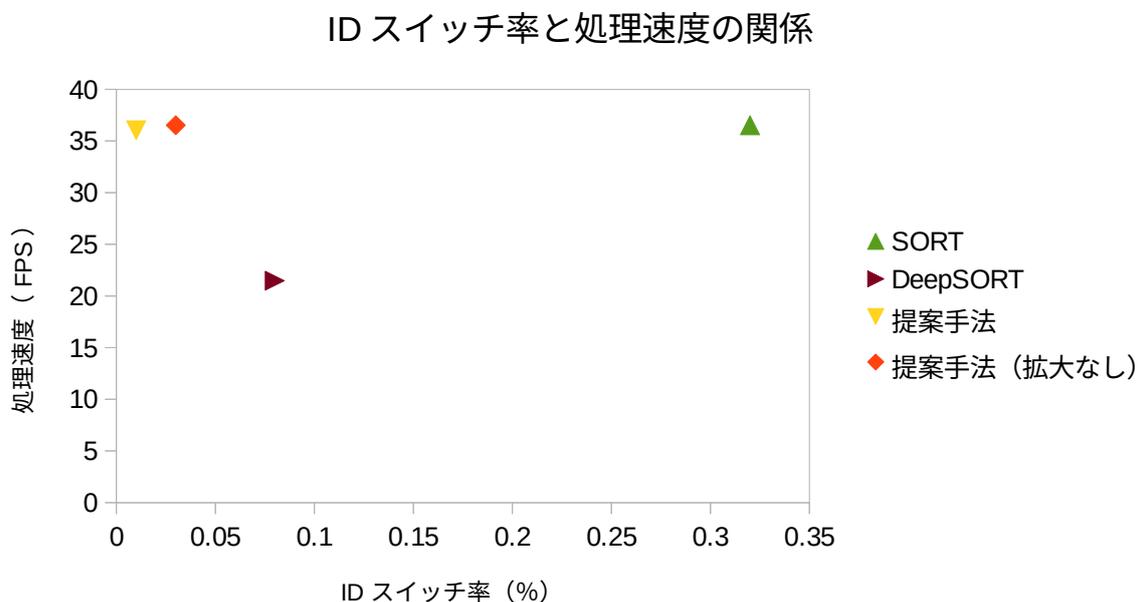


図 14: 提案手法の評価

提案手法の追跡精度と処理速度について考察する。図 14 に表 1 の結果の散布図を示す。横軸の ID スイッチが少ないほど精度が優れ、縦軸の FPS が大きいほど処理速度が早いので、図の左上に近いほど性能の良い手法と言える。図 14 には、表 2 の画像の拡大と再検出を追加する前（拡大なし）の結果も示している。

図 14 より、提案手法は画像を拡大しなくても、高精度の追跡が行える上、SORT と同じ程度高速に処理できる。精度を最大化するために画像の拡大を追加すると、速度が少し低下するが、まだリアルタイム処理に必要な 30 FPS より高速な処理が可能である。

提案手法の追跡 ID の結合では、隠れ等で一旦見失った追跡対象を再度対応づけている。DeepSORT では、外観特徴を導入することで同様の処理を行っている。しかし、DeepSORT では、外観特徴の比較処理によって、計算時間が多くかかる。この計算時間は、検出と追跡された牛の数の 2 乗に比例するので、牛の数が多い場合、計算に必要な時間が長くなり、処理速度が低下してしまう。

提案手法では、この処理に bbox 間の距離計算のみを使用している。これにより、牛の数が多くても、速度にはあまり影響せず、リアルタイムの複数牛個体追跡が行える。つまり、提案手法は簡易な処理で、SORT と同じ程度高速で DeepSORT を上回る追跡精度を達成したと言える。

## 5 おわりに

本論文では、重なり滞留が多く発生する牛の群れ中の複数牛個体を追跡するために、従来手法である SORT に対して改良を行った。重なり滞留で bbox が検出できなかった場合に、「滞留」状態を導入し、滞留を考慮した処理を追加した。「滞留」状態の軌跡に対してカルマンフィルタ予測を停止することで、予測位置が外れることを低減した。また、追跡 ID の結合により、一旦見失った追跡対象を再度対応づけられるようにした。さらに、画像の拡大と再検出により検出漏れを減らした。これらの処理を追加することで、SORT で多く発生する追跡の誤り (ID スイッチ) を減らした。

実験においては、提案手法と SORT、DeepSORT を牛の群れの動画に適用し、ID スイッチの発生回数により評価した。その結果、SORT で 48 回、DeepSORT で 12 回発生していた ID スイッチの回数を、2 回まで削減することができた。また処理速度についても、DeepSORT より速く、SORT と同程度に高速で、リアルタイム処理に必要な 30FPS を超えることを示した。このように、提案手法は SORT の高速性を維持しつつ、DeepSORT を超える追跡精度を達成した。

今後の課題としては、さらなる検出精度の向上が挙げられる。実験では提案手法でもまだ 2 回の ID スイッチが起こった。これは、物体検出により 2 頭の牛を 1 つの bbox として検出したことが大きな原因であった。物体検出器の精度を上げることができれば、追跡精度もより向上できると考えられる。

## 謝辞

本論文の作成にあたり、適切なサポート、また丁寧に指導して下さった椋木雅之教授には深く感謝いたします。相談のときは、いつも優しくアドバイスなどのご教示いただき感謝の念に堪えません。ありがとうございました。また、森 慎太郎 先輩、兒玉 光平 先輩より、貴重なご指導とご助言を賜りました。感謝申し上げます。よく声をかけてくれる熊田 陸 さん、長友 耀希 さんにも厚く御礼を申し上げ、感謝の意を表します。最後に、所属する椋木研究室のみなさまには多くのご支援をいただきました。お礼申し上げます。ありがとうございました。

## 参考文献

- [1] Bewley, Alex and Ge, Zongyuan and Ott, Lionel and Ramos, Fabio and Upcroft, Ben 『Simple Online and Realtime Tracking』 2016 IEEE International Conference on Image Processing (ICIP) (2016)
- [2] Joseph Redmon and Santosh Divvala and Ross Girshick and Ali Farhadi 『You Only Look Once: Unified, Real-Time Object Detection』 arXiv:1506.02640 (2016)
- [3] Alexey Bochkovskiy and Chien-Yao Wang and Hong-Yuan Mark Liao 『YOLOv4: Optimal Speed and Accuracy of Object Detection』 arXiv:2004.10934 (2020)
- [4] [https://qiita.com/m\\_\\_k/items/8e2cb9067ec5d720c30d](https://qiita.com/m__k/items/8e2cb9067ec5d720c30d)
- [5] <https://qiita.com/shoku-pan/items/35eae224c59989957623>
- [6] Wojke, Nicolai and Bewley, Alex and Paulus, Dietrich 『Simple Online and Realtime Tracking with a Deep Association Metric』 2017 IEEE International Conference on Image Processing (ICIP) (2017)
- [7] Dao, Trung-Kien and Le, Thi and Harle, David and Murray, Paul and Tachtatzis, Christos and Marshall, Stephen and Michie, Craig and Andonovic, I. 『Automatic Cattle Location Tracking Using Image Processing』 2015 23rd European Signal Processing Conference (EUSIPCO) (2015)
- [8] Stephenson, Mitchell and Bailey, Derek 『Do Movement Patterns of GPS-Tracked Cattle on Extensive Rangelands Suggest Independence among Individuals?』 Agriculture. 7. 58. (2017)
- [9] <https://explore.org/livecams/farm-sanctuary/cattle-pond-pasture-farm-sanctuary>
- [10] Jen-Kai Tsai, Chen-Chien Hsu, Wei-Yen Wang, Shao-Kang Huang 『Deep Learning-Based Real-Time Multiple-Person Action Recognition System』 Sensors 20, no. 17: 4758 (2020)
- [11] Mingfei Gao and Ruichi Yu and Ang Li and Vlad I. Morariu and Larry S. Davis 『Dynamic Zoom-in Network for Fast Object Detection in Large Images』 arXiv:1711.05187 (2018)
- [12] <https://ichi.pro/hi-saidai-yokusei-nms-112304374359350>