

親鳥ロボットと物体検出を用いたヒナの摂食誘導システム A robotic feed guidance system using chick detection

市原 禄朗[†]新村 毅^{††}藤波 香織^{†††}

Rokuro Ichiara Tsuyoshi Shimmura Kaori Hujinami

1 はじめに

養鶏場ではヒナの摂食状況の偏りが問題となっており、最悪の場合には栄養不足で死んでしまう個体が発生する。これを引き起こす理由の一つとして一部のヒナが積極的に餌を食べないということがあげられる。本来、親鳥はヒナに餌の位置を知らせ、餌を食べさせるために、Food-call という音を発しながら床をくちばしでつつく動作をする。ヒナはそれを見ると親鶏の周りに集まり、親鶏と同様に床をつつき始める。このようなヒナの習性は、親鶏が餌を食べていると思いつつため起こると考えられている [1]。また、ヒナは親鳥に育てられることで様々な好影響があり、摂食行動をより頻繁に行うようになることや、恐怖心が減ることがわかっている [2]。しかし、親鶏がヒナや人間を攻撃してしまうことや、それを阻止するために安全マージンをとると生産効率が悪くなってしまふという問題、そして、母子行動はホルモンに依存するため、実際に親鳥を養鶏場に導入するという事は難しい。

そこで本稿では、親鳥を模したロボット [3] (以降、親鳥ロボット) と物体検出を用いたヒナの摂食を誘導するシステム (以降、分散型摂食誘導システム) を提案し、システムの概要や使用したモジュール、そして、物体検出精度向上のためのデータ拡張や自動化の展望について述べる。システムによって、親鳥を用いずヒナの摂食状況の偏りを軽減することや、親鳥の育成効果を得ることが期待できる。

2 関連研究

2.1 親鳥がヒナに与える影響に関する研究

親鳥は Food-call と呼ばれる音を発し、ヒナの注意を引くことで摂食行動を促進するが、他にも親鳥の存在は、ヒナの行動に様々な影響を与えることが Shimmura らの研究 [2] で明らかになっている。この研究ではヒナを親鳥が飼育するグループと家畜用赤色加熱ランプで飼育するグループに分けて飼育と実験を行う。実験の結果、親鳥に育てられたヒナは食事のために行う床つつきや、砂浴びなどの能動的行動に費やす時間が有意に長くなった。その要因として、親鳥の鳴き声やエサや床をつ

つく視覚刺激によって促進されることが述べられていた。結果より、本研究ではヒナは親鳥に育てられることで摂食などの行動が促進され、恐怖心が低下することが結論付けられていた。本システムでは親鳥ロボットを用いて摂食誘導を行うため、親鳥を導入するコストや生産効率を保ったまま親鳥の好影響を得ることができる。

2.2 動物の位置検出に関する研究

動物の位置検出を行う研究は多数行われており、位置検出を行う研究の 1 つとしてセンサとの距離を用いて位置検出を行う研究がある。Quwaider らが行った研究 [4] では、動物福祉の向上と行動検出を目的とし、非ケージ環境での鶏の行動や位置を遠隔で監視することによって、行動を自動分類するための枠組みを開発した。実験では無線式センサを鶏に装着し、餌場や水場などの設備にも同様にセンサを装着することで、信号強度によって鶏の位置や、設備の利用状況を検出した。実験では鶏にセンサを取り付け、その他センサとの位置関係を算出することで、設備から 1m 以内に鶏が立ち入ったかを判定した。センサによる判定とビデオを用いた目視による判定で精度を検証したところ、84%以上の精度が示された。これより、センサによる位置特定の可能性を示した。Quwaider らの研究では、センサを用いて位置検出を行ったが、本システムではセンサを用いず位置検出を行っているため、低コストで位置の検出が行なえ、センサを取り付ける手間も削減することができる。

2.3 動物の行動誘導に関する研究

動物の行動誘導を行う研究の 1 つとして、小針らの研究 [5] が挙げられる。この研究では、ウシの嗜好性が高いカウブラシを用いて誘引をすることで、乾草槽を占有している優位牛の運動場の利用状況を変化させ、乾草槽の軋轢を改善することを試みた。実験は牛 15 頭に対して行い、乾草槽とは反対側の 2 箇所を設置し、行動を観察した。結果として、カウブラシ設置区間の利用頭数が上がったが、乾草槽の混雑状況は緩和しなかった。原因としてはカウブラシの使用が採食の欲求に対しての影響が少ないことや、設置位置が乾草槽から 20m ほどで、若干の移動を要することが乾草槽付近の軋轢改善まで至らなかった要因と考えられた。また、その他研究では換毛期に実験を行い、より多くの時間カウブラシが利用されていたため、時期もカウブラシの誘引性に影響があると考察していた。この研究はカウブラシが時期や位置などの要因から誘引性に欠けると述べられているが、本システムで使用する Food-call は、時期や位置の影響が少なく、摂食に対して本能的に誘導を行うことができる。

3 分散型摂食誘導システムの設計と実装

3.1 概要

提案する分散型摂食誘導システムは、複数の親鳥ロボットと多数のヒナがいる状況で、ロボットとヒナの位

[†] 東京農工大学 大学院 生物システム応用科学府 生物機能システム科学専攻 Department of Bio-Functions and Systems Science, Tokyo University of Agriculture and Technology

^{††} 東京農工大学 大学院 農学研究院 生物生産科学部門 Division of Science of Biological Production, Tokyo University of Agriculture and Technology

^{†††} 東京農工大学 大学院 工学研究院 先端情報科学部門 Division of Advanced Information Technology and Computer Science, Tokyo University of Agriculture and Technology

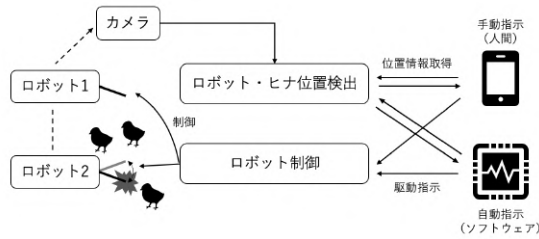


図 1: システムの概要図

置関係から任意のロボットを決定したのちに駆動させて Food-call を発生させる。このため、システムにはロボットおよびヒナの位置検出機能とロボット制御機能、制御対象決定機能が必要になる。制御対象の決定は、人間がヒナとロボットの位置関係をスマートフォンや PC 上で確認して手動で行うパターン（手動パターン）と、何らかのルールにもとづきソフトウェアが行うパターン（自動パターン）の 2 通りをサポートする。位置検出機能ならびにロボット制御機能と制御対象決定機能の間の呼び出しインターフェースを制定することで、いずれのパターンでも容易に対応できるようにする。ここで、システムの概要図を図 1 に示す。

3.2 親鳥ロボット部

親鳥ロボットは親鳥の Food-call の際に行う床をつつくような動作と注意を引くような音を出す必要がある。それを再現するために、図 2 のような床を叩く機構を持ったロボットを作成した。ロボットは Raspberry Pi によって動作しており、モータドライバによってステッピングモータを制御する。その他機構の枠部などは 3D プリンタによって作成した。また、ロボットはヒナが寄ってくるのが想定されるため、全体を覆い、先の床を叩く部分のみが露出している形となっている。床を叩くために、図 2 の機構ではステッピングモータが回転する。これによってプロペラがモータと同期して回り、プロペラが機構に取り付けられたスプリングを持ち上げる。そして、プロペラの凹部に到達すると、スプリングが落とされる。これによって先のゴム部分が床を叩き、床をつつくような動作と音を出すことでヒナの注意を引く。この機構ではプロペラ 1 回転あたり 4 回床を叩くような仕様となっている。また、ロボットは Raspberry Pi の起動と同時に cgi サーバを建てる。サーバではクエリ文字を介して動作命令を受け取り、モータの制御プログラムが

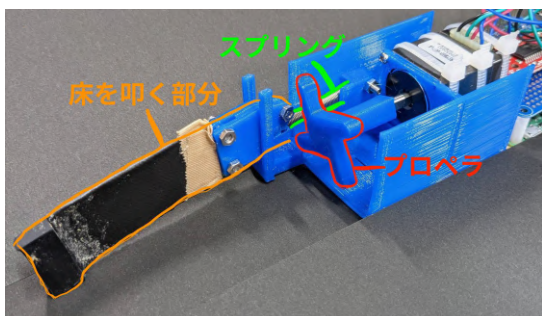


図 2: 親鳥ロボットの床を叩く機構

(赤枠: プロペラ, 黄緑枠: スプリング, 橙枠: 床を叩く部分)

実行される。命令はキュー形式で追加され、マルチスレッドで処理をすることで、命令の待機とモータの制御を非同期で行う。制御パラメータにはロボットが床を叩く速度を指定する回転速度（回転/秒）と、その動作を何秒間行うかを定める動作時間（秒）、そして動作を何セット行うかのセット数（回）、セット間の間隔を指定するセット間隔（秒）を設定可能である。パラメータのイメージを図 3 に示す。これらのパラメータは実際に親鳥が Food-call するとき様々なつつくパターンや鳴く時間、回数や間隔で行っていることから、ヒナの摂食を誘導する際に影響があると考えられたものであり、適宜変更する事によってヒナの誘導を効果的に行える。

3.3 位置検出部

位置検出部では、親鳥ロボットの検出とヒナの検出を行う。ロボットの検出には AprilTag[6] マーカを用いて行い、ヒナの検出には物体検出モデルの YOLOv5[7] を用いている。また、ロボットの位置情報は固定のため、システムの起動時に 1 回のみ行うが、ヒナの位置情報は随時変わるため、1 サイクル毎に検出が行われる。各検出方法について以下で詳しく述べる。

3.3.1 ロボットの検出

ロボットを AprilTag を用いて検出するために、本体への取り付けを行った。AprilTag はコードを読み込むことによって、そのマーカが持つ ID や上下左右の方向を得ることが可能である。ロボット本体は直方体になっていることから、上面の部分の前後に 1 枚ずつ AprilTag を貼り付け、2 枚のタグの情報から全体の矩形を取得した。この際、タグの ID が $n, n+1$ のものを同じロボットに使用することで、同一ロボットの検出を可能にした。ここで、検出の様子を図 4 に示す。検出の際は、上から撮影したときに、2 枚のタグが認識出来たときにロボット全体の矩形情報を取得し、そこから算出した中心座標と幅、高さを JSON ファイルに保存することで、インターフェースに依存せず位置検出機能からデータを読み込める形にした。JSON ファイルには位置情報の他に、認識した台数とカメラの画角情報を書き込んでいる。画角情報を組み込むことで、アプリケーションでロボット位置情報の描画をする際、キャンパスのサイズとの比率から正しく描画ができる。また、実際のロボットのサイズを元に 1cm が何ピクセルに対応するかを算出し、JSON に格納している。これによってロボットの回りにいるヒナの個体数を周囲 xcm 以内といった形で cm 指定をして取得をすることが可能となっている。

3.3.2 ヒナの検出

本システムでは、ヒナの検出を行うために YOLOv5 を使用している。YOLO は高速検出が可能な物体検出モデルであり、本システムはヒナのリアルタイムな位置情報を元にロボットを動作させる必要があるため YOLO を用いた。また、システムは、ヒナの検出データを csv ファイルと JSON ファイルに出力する。これらに格納されるヒナの位置情報は検出領域の中心 x, y 座標と幅、高さの 4 つの情報からなっており、半角スペースで結合されている。これは、YOLO のデフォルトデータ形式である。csv ファイルには検出のログを残すために、タイムスタンプと検出された位置情報を 1 羽につき 1 行出

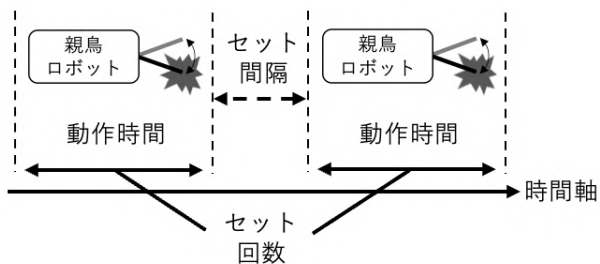


図 3: 各パラメータの説明

力し、検出毎に追記をしていく形となっている。また、JSON ファイルにはクラス名、羽数、ヒナの位置情報を出力し、検出毎に更新を行って最新の情報が格納される。制御対象決定部からリクエストが来ると JSON ファイルを返し、リアルタイムにヒナの位置情報を取得することを可能としている。しかし、ヒナが密集しやすいことや、飼育時にはおかくずなどを敷くことから背景と同化しやすいこと、かつ明るさや背景等の環境変化によって検出精度が低いという課題があった。

検出精度向上の手段として、モデル変更や学習データを増やすことが考えられた。しかし、モデルの変更はリアルタイム性を保つため難しく、学習データも入手が困難であった。そのため、データ拡張を用いて精度向上を図った。精度向上のために既存の手法 5 つと新規に提案した手法 1 つの計 6 つの手法を実装し、併用による精度向上を検証した。実装した 5 つのデータ拡張はランダムに画像の一部をくり抜く Random cropping (RC) [8]、画像をランダムに反転させる Random flipping (RF) [9]、ランダムに画像の一部に遮蔽を生成する Random Erasing (RE) [10]、オリジナル画像に物体画像を合成する手法 [11] (便宜上、以後 Synthesizing (S) と呼称)、格子状の遮蔽を生成する FenceMask (FM) [12] であり、新たに提案する手法は誤判定領域を合成する Error-synthesizing (ES) である。Error-synthesizing は誤判定領域を画像に合成する事によって、学習中に意図的に誤検出をさせる。これによって学習中に間違いやすい領域に対してのパラメータ修正、つまりは間違いを学習することが可能になると考えた。これらのすべての組み合わせに対して精度の比較・検証を行い、精度の向上を図った。

3.4 ロボット制御部

ロボット制御部では、任意のロボットの駆動指示を受け取り、該当ロボットに送信する。言語は Python で記

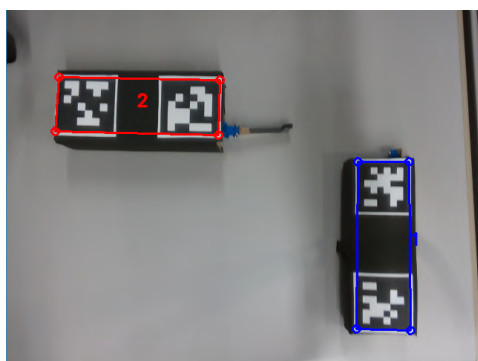


図 4: 検出の様子

述されており、REST API 形式を使用して、制御対象決定部やロボットと通信を行っている。ロボットへの駆動命令は、制御対象決定部からくる POST リクエストによって受信する。駆動指示の詳細はリクエスト時のクエリ文字を介して行われ、駆動をするロボットの ID 指定や、駆動パターンのパラメータを取得する。制御部では、リクエスト受信後、受け取った ID から駆動するロボットを決定し、ロボット側の cgi サーバへパラメータを POST し、クエリ文字経由で送る。受信したロボットは得られたパラメータから Raspberry Pi 経由でモータを制御することで任意のパターンで動作を行う。

3.5 制御対象決定部

3.5.1 概要

制御対象決定部では、位置検出部から得られたロボットとヒナの位置情報から駆動させるロボットと駆動パターンを示すパラメータを決定する。本システムは、人間がヒナとロボットの位置関係から駆動対象ロボットの決定を行う手動パターンと、何らかのルールにもとづきソフトウェアが行う自動パターンの 2 通りをサポートしている。以下で各詳細について述べる。

3.5.2 手動制御用アプリケーション

手動制御用アプリケーションは検出されたロボットとヒナの位置情報を視覚的に確認でき、得られた位置関係からロボットの駆動命令を行う。アプリケーションは Xamarin.Forms を用いて開発した。Xamarin.Forms は Windows コンソール、Android、iOS で動作するアプリを開発することができるフレームワークであるため、システムは PC やスマートフォンからの制御が可能となっている。アプリは左上にロボットの操作設定ボタンとロボットの動作停止ボタンがあり、その横に現在のロボットの操作命令の詳細が表示されている。そして、その下には位置情報取得ボタンが配置されている。最初、ボタンのテキスト表記はロボット位置情報表示となっており、押すことでロボットの位置情報を JSON 経由で取得する。その後、得られた位置情報と画面サイズから描画位置を計算し、ロボットが表示される。ロボットの描画後、ボタンの表記はヒナ位置情報表示に変わり、以後はボタンを押すごとに最新のヒナの位置情報を取得・表示をする。これらの描画は画面中央のキャンバスで行われ、ヒナが重なった場合にも判別ができるように、ヒナは枠線付きの黄色矩形で描写される。ロボットとヒナの位置情報を描画した画面を図 5 に示す。

ロボットは本体部分がボタンとなっており、駆動命令を出す際には該当ロボットの本体を押すことで現在の設定がロボットに送信され、動作を始める。また、駆動パラメータ設定は先述の左上ボタンによって行われる。ボタンを押すことで設定画面が開き、回転速度、動作時間、セット回数、セット間隔が設定できる。画面下部の更新ボタンを押すことでローカルに設定が保存され、ロボットの命令時に読み込みと送信が行われる。

3.5.3 自動制御用アプリケーション

本システムは、将来的には人を介さず、ロボットとヒナの位置関係からシステムが動作させるロボットを決定し、ロボットが駆動することでヒナの摂食を自動的に誘

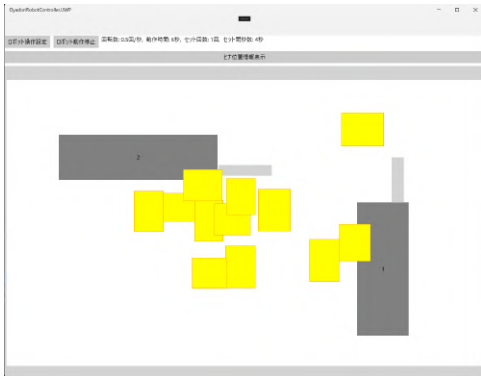


図 5: ロボットとヒナの描画をした画面

導することを目的としている。別の研究グループが個々の親鶏ロボットをエージェントと見なしたマルチエージェントによる自律分散型制御手法を開発中であるが、その前段階として、簡易的な自動化を実装した。システムは、ヒナの位置情報の分布を取得するために、検出フレームごとにロボットの中心座標から指定 cm 以内にいるヒナの羽数をロボットごとにカウントする。ロボット毎に周囲にいるヒナの羽数をカウントすることによって、どちらのロボットに分布が偏っているかを判断し、餌場にいるヒナの偏りを取得することができる。そして、指定したフレームごとに、カウンタが少ないロボットを動作させる。これによってヒナの分布状況からロボットを動かすことを実現し、自動的にヒナの摂食の偏りを緩和する仕組みとなっている。

4 実験

4.1 ヒナの検出機能

4.1.1 概要

組み合わせによる精度向上の比較・検証のために、既存の 5 つのデータ拡張と、新たに提案する 1 つのデータ拡張を実装した。これらの 6 つのデータ拡張のすべての組み合わせ、つまり 63 通りについて精度を検証・比較した。

4.1.2 使用データと精度評価指標

学習にはヒナを上から撮影した画像をデータセットとして用いた。そして、画像は条件の異なる 3 種類の画像を用意し、学習と学習中の検証には図 6 上部の単一環境の画像を 501 枚と 167 枚をそれぞれ使用した。また、精度評価に用いるテスト画像には 2 通りのデータセットを用意した。1 つ目は学習と同環境の画像 40 枚、2 つ目はそれに加えて、図 6 下部の新規環境画像を 20 枚ずつ、計 80 枚使用した。1 つ目のデータセットにより、データ拡張による精度向上について検証した。2 つ目のデータセットは、既存の学習データとは異なる新しい環境の画像をテストデータに加えている。これにより、新規環境における精度、つまり汎用性の向上を検証した。また、データ拡張はすべて 2 倍に拡張を行い、生成された画像データセットでモデルを学習し、精度を求めた。

そして、精度評価指標には、物体検出タスクの精度評価で使われる AP (Average Precision) という指標を用いた。AP とは IoU (Intersection over Union) (1) を閾値と



図 6: 学習と検証に使用した画像例 (上) とテスト用に追加した画像例 (下 2 枚)

した Precision と Recall をもとに計算する値で、検出の精度を 0-1 の数値で表すことができる。AP は縦軸に Precision、横軸に Recall をプロットした曲線を積分した値であり、これによって、各 Recall に対しての Precision の値を求められ、検出の精度と全体の正解率を含んだ計算を可能にする。本実験で用いた AP は 0.5-0.95 の IoU 閾値に対して 0.05 間隔で 10 個の AP を求め、それを平均した値を精度評価値として用いた [13]。

$$IoU = \frac{\text{正解領域と検出領域の共通部分}}{\text{正解領域と検出領域の和集合}} \quad (1)$$

4.1.3 実験結果と考察

実験では、2 通りのデータセットに対して精度を評価した。まず、単一環境のみのデータに対して上位 5 つの精度を得たものを表 1 に示す。また、異なる環境を含んだテストデータに対しての精度を同様に表 2 に示す。表中の Default とは拡張を行っていない生データに対して学習したモデルの精度を示し、○は使ったデータ拡張を示す。順位 1-5 は 63 通りの組み合わせの中で精度の高かった上位の組み合わせを示している。

評価実験の結果、表 1 のように併用を行うことで精度が向上することが確認でき、多くのデータ拡張を併用することでより精度が向上する傾向が見られた。また、表 2 のように併用を行うことで汎用性が向上し、新規環境に対しての精度が単一環境に比べて向上することが示された。しかし、反対に拡張をすることによって精度が低下する組み合わせも確認でき、それらの多くが RE, FM, ES を含んだデータ拡張であった。これらの拡張は遮蔽を生成するデータ拡張であり、併用することでヒナの小さい体をより隠してしまうことで精度の低下につながることが考えられた。

表 1, 表 2 より最も精度が向上した組み合わせは

表 1. 実験結果：単一環境

順位	RF	RC	RE	S	FM	ES	AP
Default							0.684
1	○	○		○	○		0.713
2	○	○	○		○	○	0.708
3	○		○		○	○	0.707
4	○	○		○	○	○	0.706
5	○		○	○	○	○	0.706

(RF:Random-flipping,RC:Random-cropping,RE:Random-erasing, S:Synthesizing,FM:FenceMask,ES:Error-Synthesizing)

表 2. 実験結果：異なる環境を含んだテストデータ

順位	RF	RC	RE	S	FM	ES	AP
Default							0.460
1	○	○	○		○	○	0.624
2		○	○	○	○	○	0.617
3	○	○	○	○			0.617
4		○	○		○	○	0.616
5		○				○	0.610

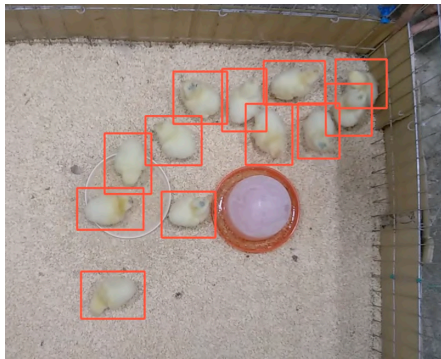


図 7: ヒナの検出の様子。重なった箇所も検出ができています。

Synthesizing 以外をすべて併用したものであることがわかった。この組み合わせを使用したデータ拡張を行ったデータに対して学習を行うことによってヒナの検出精度が向上し、密集している状態でもより正確な検出が行えることが可能になった。検出の様子を図 7 に示す。

4.2 ヒナの行動制御機能

4.2.1 実験環境

システムの動作を確認するために、実際にヒナに対して検出を行う環境でシステムを使用した。実験は 95cm×75cm×105cm のケージ内で行い、床にはヒナの体が冷えることを防ぐために、飼料としておかきずを敷いてある。また、ロボットの前方には餌場を準備し、ロボットの誘導によってエサを食べる事が可能な環境にした。カメラは上方 105cm に設置し、REALSENSE DEPTH CAMERA D435 を用いた。PC は DELL の Inspiron3268 を使用し、OS は Windows 10、RAM は 8GB であった。また、GPU は搭載しておらず、CPU は最大 3.50GHz の第 7 世代 インテル Core i5-7400 を搭載している。物体検出器には YOLOv5 の Small モデルを使用し、検出にかかる時間は 1 フレームあたり約 0.2 秒であった。ヒナは生後 1 日の採卵鶏のヒナを使用し、いずれも採卵鶏のオスを使用した。ヒナは最初中央部に放し、ロボットの行動による変化を観察した。そして、親鳥ロボットは 2 台用意し、1 台目はケージの左下から横 12cm 縦 10cm の場所に、2 台目は右上から横 12cm、縦 14cm に場所に設置した。

4.2.2 手動誘導

まず、アプリケーション経由で手動でシステムを動かすパターンの実験を行った。ヒナは 10 羽用い、ヒナの位置情報は随時アプリで確認し、それに合わせてロボットを動作させた。ロボットの動作はデフォルトを回転数 1 秒、動作時間を 7 秒、セット数 1、セットの間隔 1 とし

て動作を行ったが、ヒナの誘引状況によって変更した。システムはロボットを用いたヒナの誘導を研究している学生に使用してもらい、操作性や効果を検証した。

実験の結果、ヒナの誘引は 25 分程度で可能になった。最初は環境の変化に驚いていたのか中央に固まっていたが、徐々にロボットに興味を持ち始めた。その後、ロボットを動作し続けるとロボットに近寄っていった。引き続き動作をしていくと、ロボットの叩いている部分をつつき始め、その兆候が見られた後に餌場のエサを食べる様子が確認できた。ヒナが親鳥ロボットの動作によってエサを食べ始めた後は、ヒナがロボットに誘引されることでエサを食べる動作が見られ、システムによって摂食が誘導されていることを確認できた。

4.2.3 自動誘導

次に、ヒナとロボットの位置情報から自動的にシステムを動かすパターンを用いて実験を行った。ヒナは 7 羽を用い、検出フレームごとにロボットから 35cm 以内にいるヒナの羽数をカウントした。そして、100 フレームごとに、分布が少ないロボットを動作させた。ロボットには制御サーバが動作命令を出し、回転数 1 秒、動作時間を 7 秒、セット数 1 回で命令を出した。

実験の結果、自動的にヒナをロボットに慣れさせることが難しい事がわかった。これは、システムは 100 フレームごと (約 20 秒毎) に親鳥ロボットを 7 秒間動作させることから、ヒナの注意を十分に引く前にロボットが停止してしまい、慣れさせることが難しかったと考えられた。そのため、自動誘導実験で用いたヒナの中でロボットに慣れたと判断された 2 羽と、4.2.2 節で用いたロボットに十分慣れている 10 羽の合わせて 12 羽を用いて再実験を行った。また、ロボットの動作を 300 フレーム毎に、そして、ヒナの注意を十分に引くようにロボットの動作時間を延長し、回転数 1 秒、動作時間を 3 秒、セット数 3、セットの間隔 2 秒で命令を出した。

再実験の結果、図 8 のようにヒナが偏った際は適切にロボットが起動し、誘導することでヒナの分布を均等に分けることが出来た。また、ロボットが動作することにより、餌場のエサを食べるヒナも見られ、簡易的ではあるが、自動化の可能性を得る事ができた。そして、ヒナも適度に分散していたため、検出の精度も高く、10 羽-12 羽を安定して検出することが出来ていた。

4.3 考察

実験を通してシステムの課題点がいくつか見つかった。1 つ目はヒナが密集した際の検出精度である。2-3 羽が密集したヒナに対しての検出は行えていたが、最初にヒナを設置した際は図 9 のように密にヒナが集まってしまい、検出が 2-5 羽分しか行えなかった。しかし、これ以上の検出精度の向上は難しいと思われるため、検出領域を結合し、その面積によって分布を計算するという手法が取れると考えられた。

2 つ目はロボットの動作音である。プロペラによって機構を持ち上げる関係上、摩擦によって機械音が発生し、それによってヒナが怖がってしまう事が起こった。音が大きく出ている親鳥ロボットはテープで該当部分を覆うことで摩擦音が軽減し、ヒナが誘引することが確認できたため、今後は機械音を抑制し、叩く音の

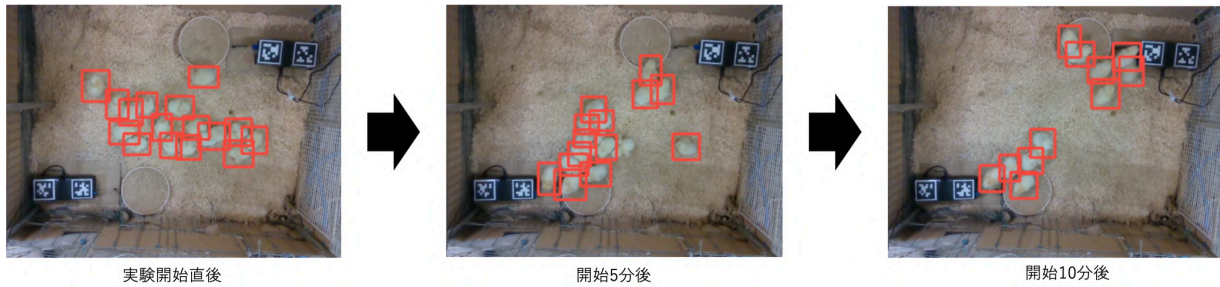


図 8: 自動化によって分布が均等になっていく様子

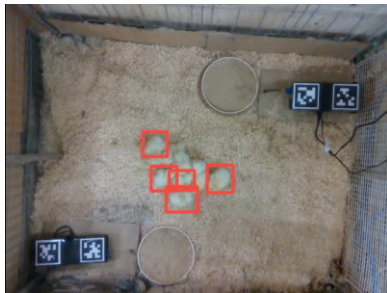


図 9: ヒナが密集した際の検出精度

みを出せるロボットの開発が必要であると考えられた。

また、自動化については可能であると考えられたが、ヒナにロボットを慣れさせるという点で課題が見られた。そのため、ヒナがロボットに興味をもち、誘引されるまでは継続的にロボットを動作させるなど柔軟にロボットの動作パターンを変更し、誘引が見られてから上記のような自動化を行うことで、より最適なシステムの作成ができると推察される。また、分布が均等になっているときに片方のロボットが動作してしまうことで一時的にヒナの分布が偏ってしまう状態も観察出来たため、分布が一定になっていると判断された際は、両方のロボットを動かすことで分布を保ったまま、ヒナの摂食を誘導できると考えられた。ロボットによるヒナの行動制御を研究している東京農工大学農学府畜産学研究室の学生にシステムを使用してもらい、以下のコメントを得た。

- 予想以上にヒナを分散できた。
- アプリでロボット動作を、特にセット回数と間隔を指定できたのが良い
- 飼育者が離れている状態を想定すると精度がもっと高いほうが良い
- 分布が均等な場合にも片方のロボットが動作してしまうことでヒナが過剰に集まり、怪我につながる可能性がある

精度に関しては面積の工夫で対処することを考えている。分布が均等である判定はカウンタに閾値を設定し、差がそれ以内ならどちらも動作させる手法が取れると考えられる。

5 おわりに

本稿では、親鳥ロボットと物体検出を用いたヒナの摂食を誘導するシステムを提案した。物体検出を用いる事によって人が介入することなしにシステムが自動的にヒナの位置情報を把握することが可能となった。また、マ

アイコンを組み込んだロボットを使用する事によって、システムからロボットを選択し、かつ任意の動作が行うことが可能になった。実験の結果、ヒナの誘引にはロボット自体の静音性や動作のパターン、そして動作する間隔が効果的な事が判明した。実験によってシステムがヒナの状態を理解した上で、ロボットを最適に動かすような自動化の展望が示され、動物とロボットの人を介さないインタラクションの可能性を提示できた。

謝辞

本稿の実験を行うにあたって、多数のご協力をいただきました。東京農工大学 農学府 畜産学研究室 山口 駿介氏に感謝いたします。

参考文献

- [1] J. Edgar, et al. Influences of maternal care on chicken welfare. *Animals*, 6(1), Article No. 2, 2016.
- [2] T. Shimmura, et al. Effect of broody hens on behaviour of chicks. *Appl. Anim. Behav. Sci.*, 126(3-4), pp. 125-133, 2010.
- [3] 安田 他. ロボット工学を用いたプロイラー雛の行動制御. *Animal Behav. Manag.*, 53(3), p. 86, 2018.
- [4] M. Quwaider, et al. Development of a wireless body-mounted sensor to monitor location and activity of laying hens in a non-cage housing system. *Trans. ASABE*, 53(5), pp. 1705-1713, 2010.
- [5] 小針 他. カウブラシに対する嗜好性を利用した運動場の利用状況の改善可能性. *動物の行動と管理学会誌*, 58(1), pp. 10-16, 2022.
- [6] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Proc. ICRA2011*, pp. 3400-3407. IEEE, 2011.
- [7] glenn-jocher Github ultralytics/yolov5/train custom data. <https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data>. Accessed: 2022-6-24.
- [8] A. Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS'12*, pp. 1097-1105, 2012.
- [9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.
- [10] Z. Zhong, et al. Random erasing data augmentation. In *Proc. AAAI-20*, pp. 13001-13008, 2020.
- [11] D. M. Montserrat, et al. Training object detection and recognition cnn models using data augmentation. *Electron. Imag.*, 2017(10), pp. 27-36, 2017.
- [12] P. Li, et al. Fencemask: A data augmentation approach for pre-extracted image features. *arXiv:2006.07877*, 2020.
- [13] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.