

2021 年度 修士論文

CowFindAR :

牛顔個体識別を用いたモバイル端末向け  
管理情報提示システム

宮崎大学大学院 工学研究科 工学専攻  
機械・情報系コース 情報システム工学分野

学籍番号 T2003258

兒玉 光平

指導教員 棕木 雅之 教授

2022 年 1 月 24 日

## 概要

本研究では、個体識別による入力作業の補助や、AR による直感的な情報の提示により牛の管理を支援する手法を提案し、提案手法に基づいたクライアントサーバシステム「CowFindAR」を実装した。

近年、農業分野における様々な問題の解決に向けて、スマート農業が注目されている。畜産農業においても、スマート農業への取り組みがなされているが、便利なものは導入コストが大きい。導入コストが低いものにも優秀なシステムはいくつか存在するが、データを参照するたびに 10 桁の数字を入力する必要があるなど、管理作業を煩雑にする問題を抱えている。

CowFindAR は、カメラ映像の取得やユーザに対する情報の提示を行うためのクライアントシステムと、深層学習を用いた画像識別を行うためのサーバシステムからなる、2 層クライアントサーバシステムである。サーバシステムは、クライアントシステムからリクエストされた画像に対して、深層学習を用いた画像識別を行い、処理の結果をクライアントシステムにレスポンスする。クライアントシステムは、取得した画像をサーバシステムへリクエストし、サーバシステムによる画像識別の結果をユーザに AR で提示する。

実験では、サーバシステムの性能について評価した。識別性能の実験では、26 頭の牛データをシステムに登録し、同一個体の牛を識別する実験を行った。実験のなかでレスポンスタイムを計測することにより、サーバシステムの速度性能についても評価した。これらの実験から、サーバシステムの識別性能と速度性能は、トレードオフの関係になっていることが分かった。

また、システムをスマートフォンに実装し、クライアントシステムの評価実験も行った。実験では、AR の重要な研究課題である、重畳する情報の位置合わせについて評価した。結果として、提案手法は比較対象となるシステムより高い精度で位置合わせを行い、AR システムとして十分な位置合わせの精度を持つことが確認された。

## 目次

1	はじめに	1
2	牛個体情報を扱う従来手法	3
2.1	牛個体情報の管理	3
2.2	牛個体識別	6
2.3	AR を用いた管理情報の提示	7
3	CowFindAR	8
3.1	システム全体の構成	8
3.2	サーバシステム	10
3.3	クライアントシステム：ステップモード	17
3.4	クライアントシステム：シームレスモード	21
4	サーバ性能の評価	24
4.1	評価実験	24
4.2	検出実験	27
4.3	識別性能	30
4.4	レスポンスタイム	31
4.5	考察	34
5	AR による位置合わせの正確性の検討	35
5.1	実験方法	35
5.2	実験結果	36
5.3	距離推定の評価	39
5.4	考察	40
6	おわりに	41
	謝辞	42
	参考文献	43

# 1 はじめに

近年、農業分野における様々な問題の解決に向けて、スマート農業が注目されている。農林水産省によれば、スマート農業とは、「ロボット技術や情報通信技術 (ICT) を活用して、省力化・精密化や高品質生産を実現する等を推進している新たな農業 [1]」である。

畜産農業においても、スマート農業への取り組みがなされている。例えば、cntxts Inc. が提供する Smart Cattle<sup>®</sup>[2] では、牛の体にセンサデバイスを取り付け、センサデバイスから取得した情報をスマートフォン等で受け取るシステムを提供している。1頭の牛につき1つのセンサデバイスを取り付けて管理する手法は、牛の個体情報や位置情報を正確に管理できるが、導入のコストが高いため多くの畜産農業従事者が利用できる手法とは言い難い。

大きなコストがかからない手法としては、株式会社ファームノートが提供するクラウド牛群管理システム Farmnote Cloud[3] が挙げられる。Farmnote Cloud は、繁殖対象牛、ワクチン接種が必要な牛などの項目での絞り込みや、アカウントを経由した獣医師など外部の関係者とのデータのやり取りなどにより詳細な牛の情報まで管理できる。しかし、耳のタグに表記された個体識別番号を手動で入力したり、文字情報のみで情報を出力したりと、不便を感じる部分も多い。

そこで本研究では、導入コストが低いモバイル端末向けのシステムとして、入力を牛顔個体識別により、出力を AR(Augmented Reality / 拡張現実) によりサポートできる、畜産動物の管理を支援するシステム CowFindAR(Cow Face individual identification AR) を提案する。システムは深層学習を用いて、牛顔画像から牛個体を識別し、個体識別番号を取得する。さらに、得られた個体識別番号を画像内の牛に重ねてユーザに提示することができる。牛個体に関する管理情報は、個体識別番号を使って管理されていることが多い。現状のシステムでは実装していないが、そうした管理情報を連携させることで、現実世界の牛を撮影した画像上に、その牛に関する情報を重畳表示することができる。これにより、例えば、病気を持つ牛を畜産農業従事者に分かりやすく提示することができる。畜産農業従事者の間では、牛白血病など感染症の病原体を持つ牛の管理が問題となっている。感染症の種類によっては、病原体を持つ個体を一目で見分けるのが困難であり、管理が難しいケースもしばしば存在する。AR を用いて現実世界に情報を重畳する形で強調して提示することで、畜産農業従事者に直感的に注意を促すことができる。

以下、第2章では牛個体情報を扱う従来研究について議論する。第3章では提起した問題を解決するための提案手法である CowFindAR について説明する。第4章ではシステ

ムの基本的な性能について評価し、第 5 章では AR による位置合わせ精度を評価する実験を行う。第 6 章では研究の内容をまとめるとともに、今後の展望について述べる。

## 2 牛個体情報を扱う従来手法

### 2.1 牛個体情報の管理

農林水産省によると、牛には個体識別番号が書かれた耳標 (図 2.1) を装着する必要がある [4]。個体識別番号とは牛トレーサビリティ法に基づき国内で飼育される牛に対して割り振られている 10 桁の番号である。牛個体情報を特別なセンサデバイスを用いずに管理するシステムとして、個体識別番号を用いた種々のシステムが存在する。

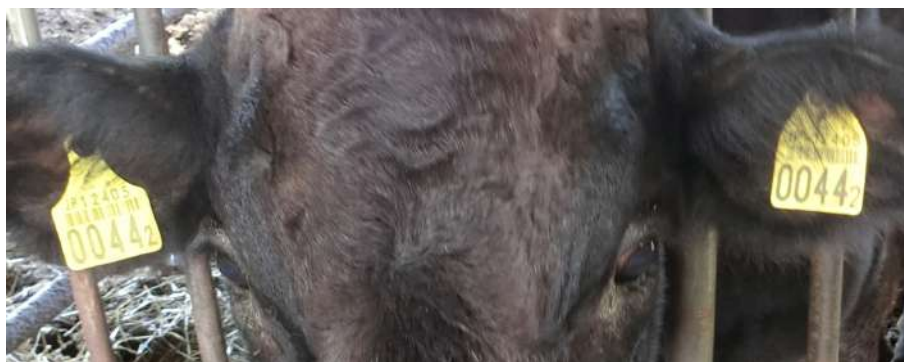


図 2.1: 耳標

独立行政法人家畜改良センターが提供する牛の個体識別情報検索サービス [5] は、10 桁の個体識別番号を入力すると、該当する牛の管理情報を提示する (図 2.2)。システムは、誕生日などを含む個体情報や管理している畜産農業施設の情報などを提示することで、牛のトレーサビリティを向上させ、誕生から出荷までの流れを辿ることができる。

株式会社ファームノートが提供するクラウド牛群管理システム Farmnote Cloud[3] は、図 2.3 に示すような繁殖、治療、牛群移動といった様々な牛の活動を登録する機能、特定の牛がこれまでにどのような活動を行ったか確認できる「ストーリー」という機能や、特定の活動を行っていることを条件に牛を絞り込んで、特定の病気に罹っている個体を一覧できる機能などを使うことができる。

一方で、Farmnote Cloud では牛の登録をする際には個体識別番号などの情報を手動で入力する必要がある (図 2.4)。牛の登録や情報の参照のために毎回 10 桁の個体識別番号を入力するのは煩雑な作業である。また、端末上に出力した情報を現実の牛と照らし合わせて閲覧する際には、文字情報を並べて出力する (図 2.5)。実際の牛と情報を見比べる場合、耳標から個体識別番号を確認する必要がある。



図 2.2: 牛の個体識別情報検索サービス [5] の入力画面 (上) と出力画面 (下)

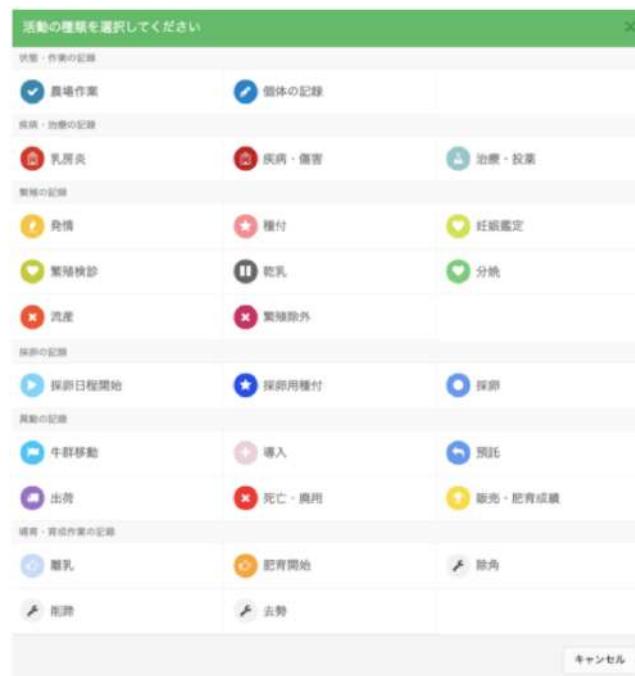


図 2.3: Farmnote Cloud[3] の活動登録画面



ここで「個体識別番号」を入力すると

図 2.4: Farmnote Cloud[3] の登録の説明

- 11-2.個体の編集の方法は？
- 11-3.個体の削除
- 11-4.個体の検索
- 11-5.個体の詳細情報
- 11-6.個体乳の検認方法は
- 11-7.給餌機能とは
- 11-8.タグ機能
- 11-9.採卵オプションご利用時の個体詳細追加表示

活動登録を進めると個体に各種情報が集約されます。個体の詳細情報を確認したい際にこちらの機能をお使いください。  
また、繁殖・疾病履歴や検定データを把握したい時、獣医さんやコンサルタントの方と個体データを共有したい際にもご活用ください。



#### 項目について

- ① 耳標番号・個体識別番号・タイプ・性別が表示されます
- ② 耳標番号と一つ前、一つ後ろの牛を選択するボタンが表示されます
- ③ ステータスが表示されます (※1)
- ④ ページを切り替えることができます (データ・繁殖・疾病履歴・検定・給餌・活動)
- ⑤ ④で選択した情報が表示されます

※1 採卵期間中の場合は採卵ステータスが追加され「採卵中」が表示されます。採卵機能についてはこちらをご覧ください。

図 2.5: Farmnote Cloud[3] の情報閲覧画面



## 2.2 牛個体識別

Janina ら [6] の研究では、スマートフォンのカメラで取得した映像から耳標を検出することでそこに牛がいることを認識し、マーカーに応じた情報を牛が映っている画面に表示することでユーザに情報を提示している。この研究の実験結果によると、耳のタグを検出できないことが多かった。汚れやオクルージョンの発生、あるいはマーカーとなるタグをあまり大きく写せないためである。

Thi Thi Zin ら [7] による研究では、画像中に含まれる耳標から個体識別番号を画像処理により文字認識することで、牛個体を識別する取り組みがされている。耳標から文字認識した結果の正確性から、92.5% の確率で牛個体の識別ができた。しかし、オクルージョンが発生している画像や品質に問題のある画像のような、正しい文字認識を期待できない耳標画像は文字認識の対象から排除している。また、耳標が外れる、隠れる、汚れる場合は実際の畜産農業施設でも多く発生する問題であるため、この手法で識別できない牛も少なくない。

高宗 [8] による研究では、様々な特徴抽出器などを用いて牛顔画像から個体識別を行い、特徴抽出器の有効性を評価する実験が行われた。実験は人の顔を学習したモデルである FaceNet と VGGFace、一般物体認識用のデータセットで学習された VGG16、VGG19、ResNet50 をそれぞれ用いて牛顔画像から特徴抽出を行い、精度を比較するというものである。結果として、識別の精度が最も高い VGG16 では、27 頭の肉牛に対して 92.8%、24 頭の乳牛に対して 97.5% の高い精度を示し、牛顔画像を用いた個体識別の可能性が示された。

牛顔はモバイル端末を用いて撮影する物体として十分な大きさを持っているため、VGG16 による特徴抽出を用いた個体識別機能をシステムに組み込むことで、個体識別番号の入力を効率的に行うことができると考えられる。

## 2.3 AR を用いた管理情報の提示

AR とは、現実空間に対して仮想的な情報を重ね合わせることで、現実を拡張した世界を作り出す技術である。現実空間に情報を重ね合わせることで、より直感的な情報の提示が可能である。代表的な AR の活用例として、IKEA Place[9] やポケモン GO[10] などが挙げられる。

前述した Janina ら [6] の研究は、AR を用いるのものであると位置づけているが、AR の重要な課題である位置合わせについては取り上げられておらず、画面いっぱいに情報を表示するのみである。AR における位置合わせとは、現実中存在する情報を付与する対象となる物体に重ね合わせる仮想的な情報を、正確な位置に配置する技術を指す。この技術を用いることで、あたかも拡張すべき現実の物体に対して情報が固定されたように見える。

多くの AR システムでは、カメラで撮影した三次元空間から床や壁などの平面を検出し、検出した平面に重畳する情報を配置するマーカーレス型の手法が用いられているが、牛顔は平面上に存在しないためこの手法を用いるのは難しい。他の手法として、あらかじめシステムに登録された特定のマーカーを検出し、検出した位置に重畳する情報を配置するマーカー型の手法がある。この手法は、マーカーさえあればどこにでも情報を配置することができる。画像中から牛顔を検出できれば、特定のマーカーを牛顔で代替することで、牛顔の位置に情報を重畳表示できる。

## 3 CowFindAR

### 3.1 システム全体の構成

本論文では、牛の管理作業を支援する手法として、牛顔個体識別と AR の利用を提案する。また、提案手法を実装したモバイル端末向け管理情報提示システムである CowFindAR(Cow Face individual identification AR)(図 3.1) を構築し、評価実験などを行う。



図 3.1: CowFindAR のホーム画面

CowFindAR は、カメラ映像の取得やユーザに対する情報の提示を行うためのクライアントシステムと、深層学習を用いた画像識別を行うためのサーバシステムからなる、2層クライアントサーバシステムである。全体の基本的な処理手順を以下に示す。

- ① クライアントシステムがモバイル端末のカメラから画像を取得する
- ② 取得した画像と共に処理のリクエストを送信する (クライアント → サーバ)
- ③ サーバシステムが画像識別を行う
- ④ 処理結果と共にレスポンスを返す (サーバ → クライアント)
- ⑤ 処理結果を用いてユーザに情報を提示する

システム全体のシーケンス図を図 3.2 に示す。

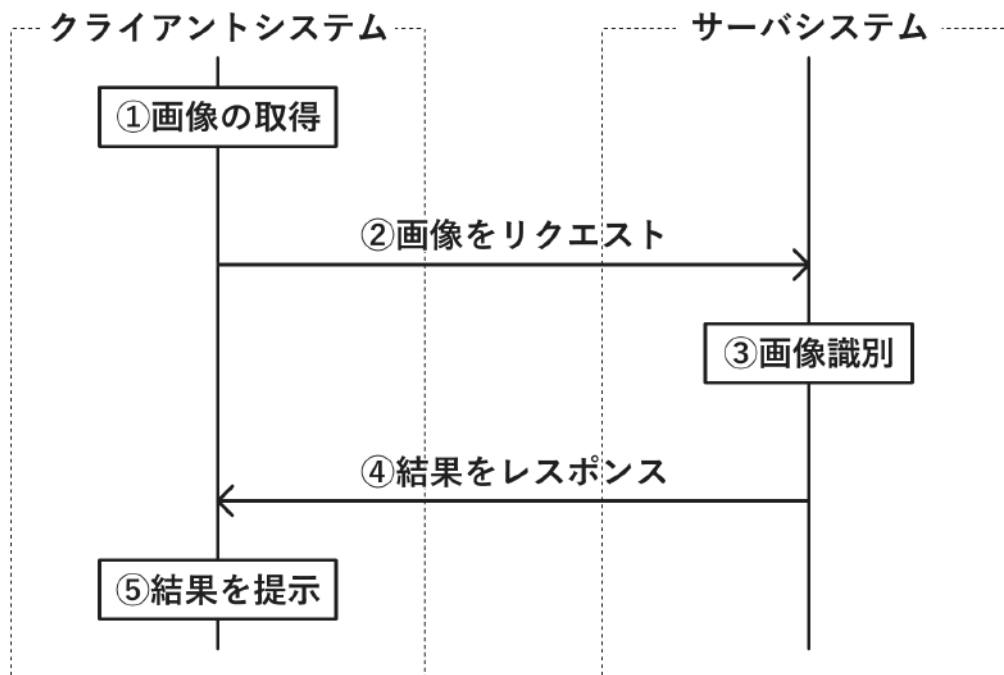


図 3.2: システム全体のシーケンス図

サーバシステムは、クライアントシステムからリクエストされた画像に対して、深層学習を用いた画像識別を行い、処理の結果をクライアントシステムにレスポンスする。牛顔画像から牛個体を識別する識別機能と、牛顔画像の牛個体を識別の候補に追加する登録機能の2機能を持ち、クライアントシステムからのリクエストによって使い分ける。

クライアントシステムは、取得した画像をサーバシステムへリクエストし、サーバシステムによる画像識別の結果をユーザに提示する。識別結果を牛顔に重畳表示するARシステムのシームレスモードと、サーバの登録機能を利用するためのステップモードを持ち、ユーザの目的によって使い分けることができる。

クライアントシステムとサーバシステム間の通信は、HTTPでやり取りされる。HTTPは多くの場合WebブラウザとWebサーバのやり取りに使われる通信プロトコルで、クライアントシステムのリクエストに対してサーバシステムのレスポンスでデータのやり取り

りを行う。HTTP によるリクエストにはいくつかの種類が存在するが、今回はクライアントシステムからサーバシステムへ画像を送る必要があるため POST によりリクエストする。

画像識別は深層学習を用いた比較的計算量のかかる処理である。クライアントとなるモバイル端末に負担をかけないように、画像識別は計算資源を導入しやすいサーバシステムで処理する。一方この場合、クライアントからサーバに撮影画像を送信する必要がある。これによる通信量の増大とレスポンスの低下が懸念される。応答時間については、後述のシステム評価の際に評価する。

## 3.2 サーバシステム

サーバシステムは、クライアントシステムからのリクエストと画像を受け取って画像識別を行う役割を持つ。リクエストによって登録機能と識別機能の 2 機能を使い分ける。

登録機能では、牛個体を識別の候補に追加する。リクエストされた画像から牛顔の検出を行い、牛顔画像を切り出す。切り出した牛顔画像から特徴ベクトルを抽出し、個体識別番号とともにデータベースに登録する。

識別機能では、牛個体を識別する。リクエストされた画像から牛顔の検出を行い、牛顔画像を切り出すとともに、牛顔領域を示す座標を取得する。切り出した牛顔画像から特徴ベクトルを抽出し、登録済みの牛個体と比較する。最も類似している牛個体の個体識別番号と切り出した牛顔領域を示す座標をレスポンスする。

サーバシステムは、Python と Flask で構築している。Python は高い汎用性と拡張性を持つプログラミング言語で、様々な用途にわたって利用されている。Flask は、Python 向け Web アプリケーションフレームワークで、他のフレームワークと比較して導入が容易で、学習コストも低いことが特徴である。

### 3.2.1 牛個体登録機能

登録機能は、牛個体を識別の候補に追加するために、リクエストされた画像から抽出した牛顔画像の特徴ベクトルをデータベースに追加する機能である。牛個体登録機能は以下に示す 2 段階のリクエストを受け付ける。

- ① リクエストとして画像を受け取り、レスポンスとして牛顔検出の成否を返す。
- ② リクエストとして個体識別番号を受け取り、レスポンスとして登録の成否を返す。

第1段階では、リクエストされた画像中に牛顔切り出しを利用して牛顔が含まれるか検出する。牛顔を検出した場合は、登録リクエストの第2段階で特徴ベクトルの抽出に用いるために、牛顔画像を切り出して保持し、検出成功メッセージをレスポンスする。牛顔が検出されなかった場合は、検出失敗メッセージをレスポンスする。

第2段階では、リクエストとして画像ではなく個体識別番号を受け取る。リクエストを受けたサーバは保持した画像から特徴ベクトルを抽出し、リクエストされた個体識別番号と共にデータベースに追加する。

データベースの1レコードには、牛個体1件分のデータである個体識別番号と4096次元の特徴ベクトルを含む。同じ牛個体について、複数のレコードを登録することで、複数枚の牛顔画像データを保持できる。データベースにはCSVファイルを用いる。

牛個体登録時のサーバのプロセスを表したDFD(Data Flow Diagram)を図3.3に示す。

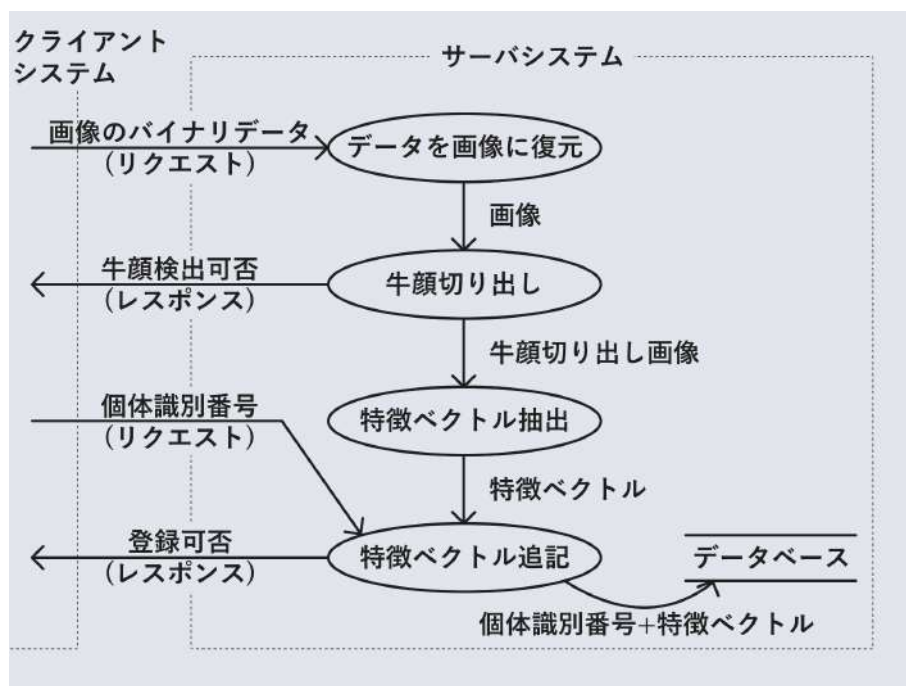


図 3.3: 牛個体登録時の DFD

### 3.2.2 牛個体識別機能

識別機能は、牛個体を識別するために、リクエストされた画像の特徴ベクトルとデータベースの特徴ベクトルを比較する機能である。牛個体識別は以下の手順で処理される。

- ① リクエストされた画像から牛顔を切り出す
- ② 得られた牛顔切り出し画像から特徴ベクトルを抽出する
- ③ 牛個体を識別する
- ④ 識別結果と検出座標をレスポンスする

牛顔切り出しに失敗した場合は、検出失敗メッセージをレスポンスする。牛顔切り出しに成功した場合は、牛顔切り出し画像から特徴ベクトルを抽出する。抽出された特徴ベクトルを SVM(Support Vector Machine) によってデータベースに記録されたデータと比較し、牛個体の識別を行う。識別結果は個体識別番号として出力し、牛顔検出座標と共にクライアントシステムへレスポンスされる。

牛個体識別時のサーバーのプロセスを表した DFD を図 3.4 に示す。

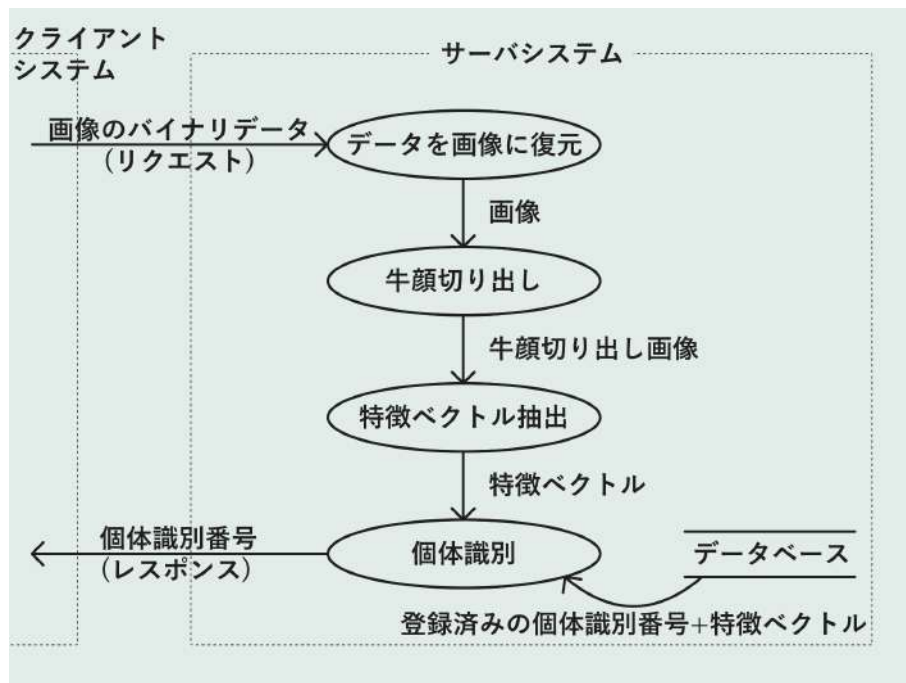


図 3.4: 牛個体識別時の DFD

### 3.2.3 牛顔切り出し

牛顔切り出しは特徴ベクトル抽出に向けた前処理である。リクエストされた画像を受け取り、その画像内の牛の顔を囲む矩形領域の上下左右を示す4つの検出座標を返す。特徴ベクトル抽出にあたってリクエストされた画像から不要な背景を取り除くために実行される。切り出し前の画像を図 3.5、切り出し後の画像を図 3.6 に示す。



図 3.5: 牛顔切り出し前の画像



図 3.6: 牛顔切り出し後の画像

牛顔切り出しを行うために、リクエストされた画像中から牛顔が含まれる矩形領域を検出する。矩形領域の検出は YOLOv5[11] によって行う。YOLO[12] は高速かつ正確な物体検出アルゴリズムの一つである。YOLOv5 はフレームワークとして Python 向けオープンソース機械学習ライブラリである PyTorch を使っている YOLO の最新版である。

検出のためのスタンダードな学習済みモデルでは牛顔を検出できないため、オリジナルのデータセットと事前学習済みモデルを用いて、牛顔検出のための学習済みモデルを作成する。データセットには過去の研究 [13] に用いた「牛全体」と「牛顔」の2クラスでラベル付けした 5921 枚の画像を含むデータセットを用いる。クラス数に対してデータが豊富にある点や AR のためのリアルタイム処理に向けて高速であることが重要である点を鑑



みて、事前学習済みモデルには高速な「YOLOv5s」を用いる。学習は 32 のバッチサイズで 50 エポック行った。

検出時は正確な検出のために、YOLOv5 が検出した結果に信頼度での閾値を設ける。閾値は 0.9 に設定し、0.9 を下回る信頼度の検出結果は無効として破棄する。画像中に複数体の牛顔が含まれる場合は、YOLOv5 が最初に検出した牛顔を採用する。

牛顔を検出した場合は、次の処理に向けて上下左右を示す 4 つの検出座標を返す。牛顔を検出できなかった場合は、空の値 (None) を返す。

### 3.2.4 特徴ベクトル抽出

特徴ベクトルは牛顔切り出し画像から抽出され、特徴ベクトル同士を比較して牛の個体を識別するために使われる。高宗 [8] によれば、牛個体識別に向けた特徴ベクトルの抽出には VGG16 が有効であるとされているため、提案手法においても VGG16 を採用する。

VGG16[14] は、表 3.1 に示す CNN(Convolutional Neural Network / 畳み込みニューラルネットワーク) で、13 層の畳み込み層と 3 層の全結合層を持つ。conv は畳み込み層、maxpool は最大値プーリング層、fc は全結合層をそれぞれ表している。

VGG16 は物体認識を目的として開発された CNN であるため、特徴ベクトル抽出器として用いるためには、ネットワークの下層を取り除くことで処理を中断する必要がある。高宗の研究では全結合層をすべて取り除き、512 次元の特徴量を出力していたが、本システムでは 3 層の全結合層のうち最後の 1 層のみを取り除くことで 4096 次元の特徴ベクトルを抽出する。

VGG16 に入力する画像サイズは  $224 \times 224 \times 3$  であるため、画像をネットワークに入力する前に、リサイズなどの前処理を行う必要がある。しかし、入力する牛顔切り出し画像は縦に長い画像であるため、正方形にリサイズした場合、情報の損失が大きいと考えられる。高宗の研究では、ヒートマップを用いて VGG16 が特徴として重要であると捉えている領域を分析する実験も行っている。実験によると、VGG16 が重要であると捉えている領域は画像の上半分に存在することが多かった (図 3.7)。

そこで本システムでは、リサイズの前に牛顔切り出し画像から上半分をさらに切り出すことで、正方形に近い形状に整形し、重要な情報の損失を避ける処理を追加した。これらの前処理の結果、図 3.8 に示すような画像から特徴ベクトルを抽出することになる。

表 3.1: VGG16[14] のネットワーク構成 (D)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

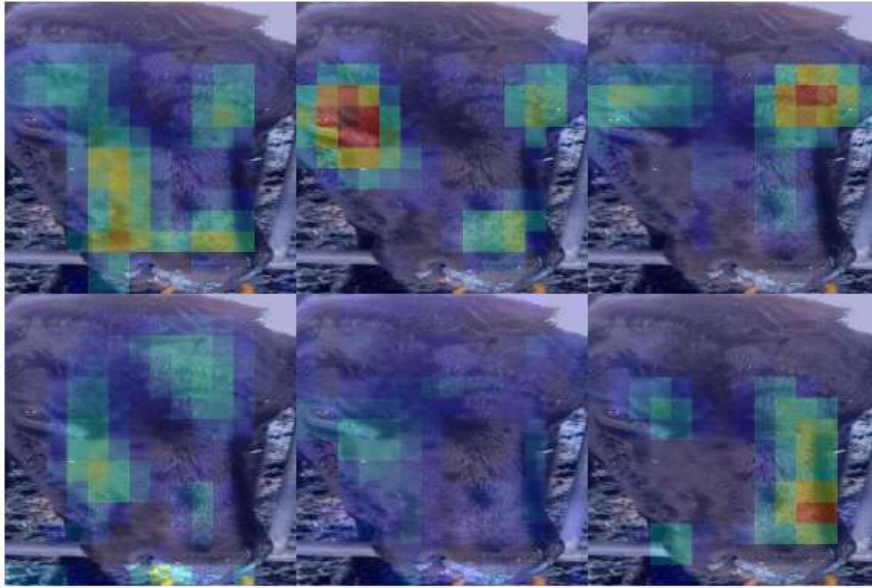


図 3.7: 高宗 [8] によるヒートマップ



図 3.8: 前処理後の画像

### 3.3 クライアントシステム：ステップモード

クライアントシステムは、取得した画像をサーバシステムへリクエストし、サーバシステムによる画像識別の結果をユーザに提示する。

クライアントシステムには、ステップモードとシームレスモードの2つのモードが存在する。ステップモードはクライアントシステムの基本動作である、カメラから画像を取得する処理、画像と共にサーバに対してリクエストを送る処理、レスポンスを受けて結果をユーザに提示する処理を段階的に行うモードである。ステップモードを起動した画面を図3.9に示す。「HOME」と書かれたボタンはホーム画面に戻るためのボタン、「C」と書かれたボタンは撮影ボタン、「登録」の文字が添えられたチェックボックスはステップモードが持つ2つの機能を使い分けるためのボタンである。



(a) 識別機能を利用する場合



(b) 登録機能を利用する場合

図 3.9: ステップモードの撮影画面

ステップモードは登録機能と識別機能の2機能を持つ。撮影ボタンを押す前にどちらの機能を使うか選択する。撮影ボタンを押すと、その時点でのカメラ映像を画像として取得し、画像と共にサーバへのリクエストを行う。使う機能によってリクエストは2種類に大別される。リクエストした画像中の牛から個体識別番号を判定してもらう識別リクエ

ストと、画像中の牛を判定の候補に追加してもらう登録リクエストである。出力は機能によって若干異なるが、基本的には画面に個体識別番号と牛の画像が静止した状態で出力される。ステップモードの状態遷移図を図 3.10 に示す。

クライアントシステムは、Unity を用いてスマートフォンに実装した。

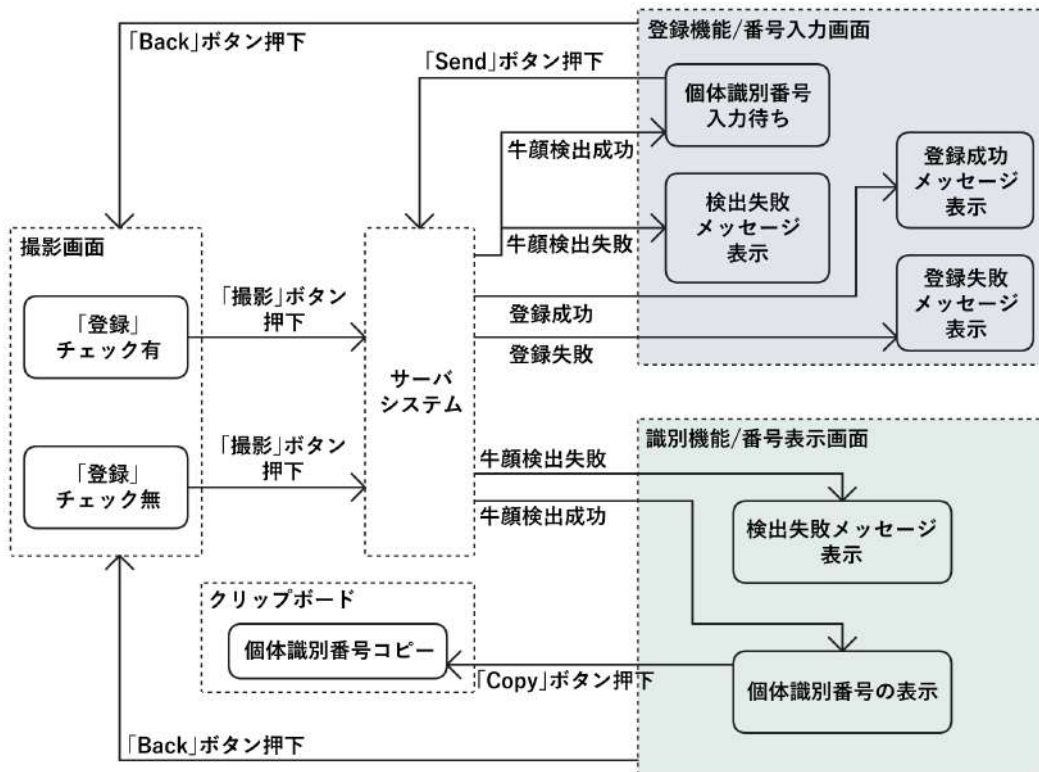
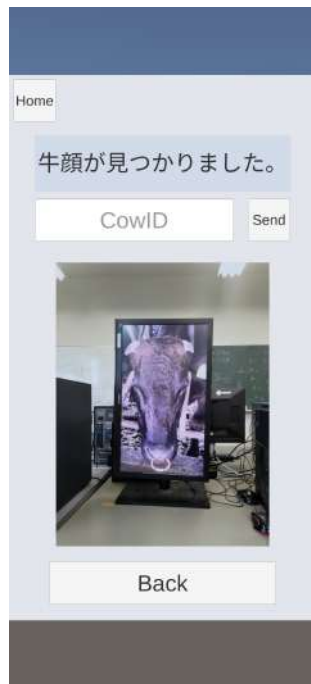


図 3.10: ステップモードの状態遷移図

### 3.3.1 登録機能

「登録」をチェックした状態で撮影ボタンを押すと、登録機能を使うことができる。登録機能は2段階のリクエストを行う。第1段階ではカメラから取得した画像をリクエストに含めてサーバに送信し、牛顔の検出可否についてレスポンスを受け取る(図 3.11)。検出に成功した場合のみ第2段階のリクエストを行うことができる。第2段階では、ユーザが、検出された牛顔に対応する牛の個体識別番号を入力する。その後、「Send」ボタンを押すと、入力した個体識別番号をリクエストと共にサーバへ送信し、登録の成否を表示する(図 3.12)。



(a) 牛顔を検出できた場合



(b) 牛顔を検出できなかった場合

図 3.11: 登録機能リクエスト (第 1 段階) 後の画面

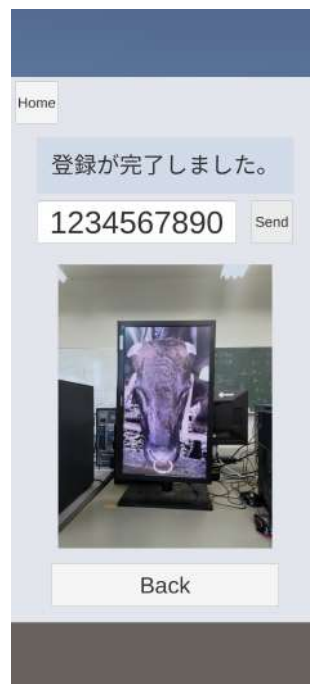


図 3.12: 登録機能リクエスト (第 2 段階) 後の画面 (登録成功)

### 3.3.2 識別機能

「登録」をチェックせずに撮影ボタンを押すと、識別機能を使うことができる。カメラから取得した画像をリクエストし、レスポンスとして識別結果である個体識別番号を受け取る (図 3.13)。



図 3.13: 識別機能リクエスト後の画面

サーバシステムで牛顔を検出できた場合、登録済みの牛の中で、最も類似した牛の個体識別番号がレスポンスされる。個体識別番号は画面上部に表示される。右の「Copy」ボタンを押すことでクリップボードにコピーできるため、既存のシステムと CowFindAR を組み合わせて使うことができる。

牛顔を検出できなかった場合、クライアントシステムでは図 3.13(b) のように表示される。

### 3.4 クライアントシステム：シームレスモード

シームレスモードはクライアントシステムの基本動作である、カメラから画像を取得する処理、画像と共にサーバに対してリクエストを送る処理、レスポンスを受けて結果をユーザに提示する処理を連続的に行うモードである。シームレスモードの動作画面を図 3.14 に示す。



図 3.14: シームレスモードの動作画面

シームレスモードはリアルタイム処理を必要とする AR のモードで、撮影ボタンや登録チェックは存在せず、識別機能のみを利用できる。前のリクエストに対するレスポンスが返ってくると自動的に次のリクエストを行うことにより、リアルタイム処理を実現している。

牛の個体識別番号の情報は、牛顔の位置に合わせて表示される。牛顔切り出しは自動で行われるが、その間隔は数秒程度となる。画像上の牛顔領域の位置に情報を固定して表示するのでは、この間隔の間にモバイル端末を動かすと、取得した映像上の牛顔の位置と情



報の表示位置がずれてしまう。そこで、現実世界に重畳させた3次元の仮想空間の「画像平面」に、情報を配置して表示する。

「画像平面」とは、現実空間において撮影した画像に相当する平面領域である。画像平面の位置は、クライアントシステムを実装するモバイル端末の位置を原点とした3次元極座標  $(r, \theta_x, \theta_y)$  によって表現される。角度  $\theta_x$  および  $\theta_y$  はモバイル端末の傾きとして、ジャイロセンサから取得する。

牛顔からモバイル端末までの距離  $r$  は、牛顔検出領域のサイズから推定する。まず、牛顔の縦幅  $CF_h[m]$ 、検出領域の縦幅を  $DS_h[px]$  から、式 (3.1) により撮影画像の1画素における現実空間の長さ  $LPP(\text{Length Per Pixel})[m/px]$  を推定する。

$$LPP = \frac{CF_h}{DS_h} \quad (3.1)$$

次に、 $LPP[m/px]$  とモバイル端末の画面サイズ (縦)  $SS_h[px]$  から、式 (3.2) により現実空間における画像平面の縦幅  $PP_h[m]$  を推定する。

$$PP_h = LPP \times SS_h \quad (3.2)$$

そして、 $PP_h[m]$  とカメラの視野角  $\theta_{fov}$  から、式 (3.3) により牛顔とモバイル端末の距離  $r$  を推定する。

$$r = \frac{PP_h}{2} \times \frac{1}{\tan\left(\frac{\theta_{fov}}{2}\right)} \quad (3.3)$$

それぞれの値についてまとめたものを図 3.15 に示す。

AR による出力は ARFoundation[15] で実現する。Unity では三次元空間にオブジェクトを配置するが、ARFoundation を用いることで、オブジェクトを配置している三次元空間を現実空間に投影し、現実空間に対して仮想的な情報を付与することができる。

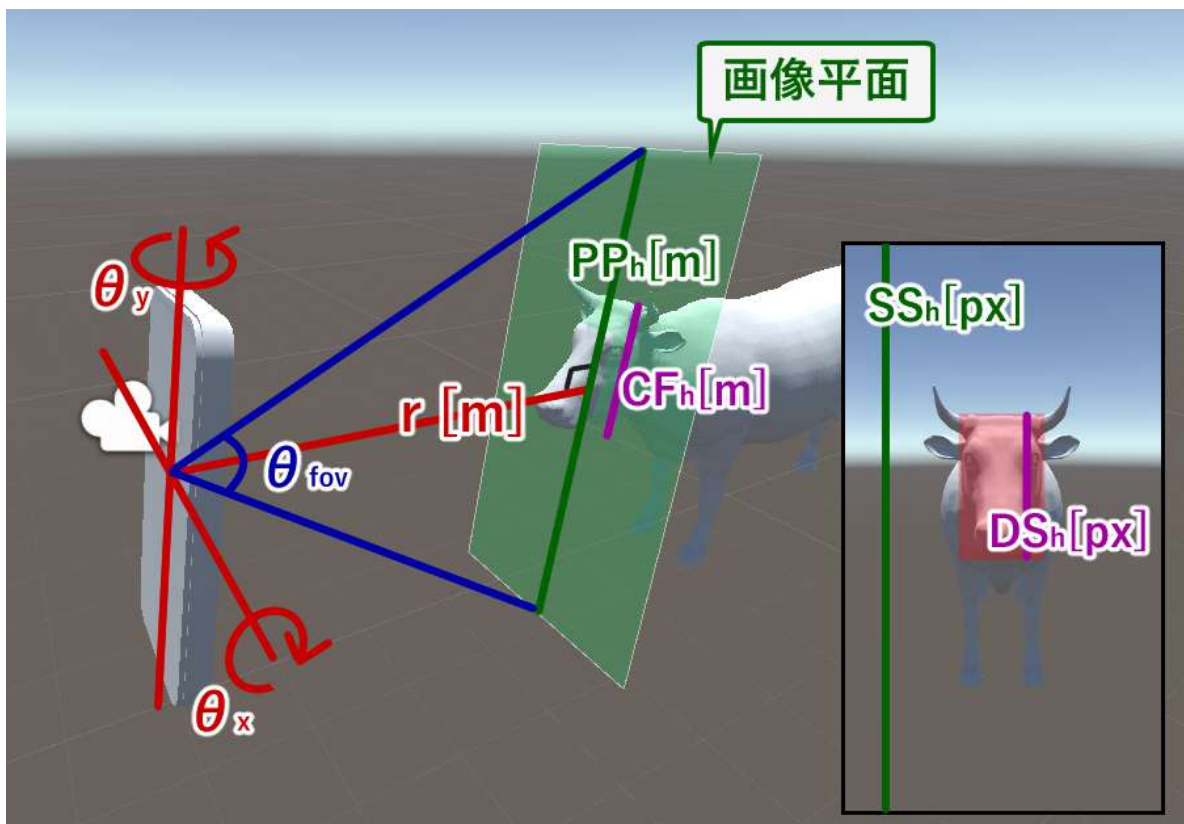


図 3.15: 「画像平面」の位置関係

## 4 サーバ性能の評価

### 4.1 評価実験

作成したシステムの使用感を確認するために、まずサーバ性能の評価実験を行う。

サーバ性能の評価実験では、システムの精度と速度を評価する。一連のシナリオを設定し、牛顔検出と個体識別の性能、サーバのレスポンスタイム、機能ごとの処理速度について測定し、評価する。評価実験に用いるクライアントシステムは、Android スマートフォン Zenfone8[16] に実装した。実験では、クライアントシステムはステップモードを使用する。

評価実験には、高宗 [8] の研究で使われた画像データの一部を利用する。高宗の画像データは 27 頭の肉牛に関して、それぞれ 10 枚の牛顔画像を含んでいる。本研究では、27 頭の牛に関して、登録用に 1 枚、識別用に 1 枚を重複することなく取り出し、紙にプリントアウトしたものをスマートフォンで撮影する。牛顔画像には 00~26 の番号を割り当てる。これらを個体識別番号とする。

登録に用いた画像を図 4.1、識別に用いた画像を図 4.2 に示す。左下に赤色で示した画像が登録用、青色で示した画像が識別用である。これらの画像をそれぞれ呼称する際は、登録用の牛には R、識別用の牛には C に 00~26 の番号を連結したものを利用する。例えば、登録用の 7 番目の牛顔画像は「R07」と呼称する。

評価実験のシナリオを以下に示す。

- ① 登録用の牛顔画像に関して、すべてを 1 回ずつ登録する。
- ② 識別用の牛顔画像に関して、すべてを 1 回ずつ識別する。



図 4.1: 登録用の牛顔画像



図 4.2: 識別用の牛顔画像

## 4.2 検出実験

実験での撮影は、1枚の画像に関してそれぞれ牛顔の検出に成功するまで行う。正面からの撮影を2回行っても検出しない場合は、牛顔画像をやや見下ろす形での撮影を2回まで行い、それでも検出しない場合は様々な角度での撮影を試す。登録時の検出結果を表4.1に、識別時の検出結果を表4.2に示す。

検出性能の評価は、どのような特徴や特性を持った画像の検出が困難であるかという観点で行う。

54枚の画像(54頭の牛)のうち、48枚については正面からの撮影2回以内に、牛顔検出に成功した。

R09とR11は、正面とやや見下ろしでは検出できなかった。R09とR11を図4.3に示す。R09に関しては、大きく見下ろす形で撮影すると検出に成功し、R11は逆に見上げる形で撮影すると検出に成功した。同じ個体の異なる牛顔画像であるC09とC11(図4.4)に関しては、正面での撮影による検出には2回とも失敗したが、やや見下ろす形での撮影には1回で成功した。

R13(図4.5)に関しては、正面と見下ろしでの検出に失敗した後に、より明るい環境やより暗い環境に画像を置く、色調補正など手を尽くしたが牛顔が検出されることはなかった。

R09、C09、R13に関しては、牛顔が暗く映っていることが原因として考えられる。逆光も影響していると考えられるが、R02やR12など、逆光が発生している画像は他にもあるため、逆光は直接的な原因ではなく、逆光により牛顔に含まれる情報のほとんどが欠落していることが原因であると考えられる。

表 4.1: 登録時の検出結果

個体番号	00	01	02	03	04	05	06	07	08	09	10	11	12	13
正面1	o	o	o	o	o	x	o	o	o	x	o	x	o	x
正面2	-	-	-	-	-	o	-	-	-	x	-	x	-	x
見下ろし1	-	-	-	-	-	-	-	-	-	x	-	x	-	x
見下ろし2	-	-	-	-	-	-	-	-	-	x	-	x	-	x
様々な角度	-	-	-	-	-	-	-	-	-	o	-	o	-	x

個体番号	14	15	16	17	18	19	20	21	22	23	24	25	26
正面1	o	o	o	o	x	o	o	o	o	o	o	o	o
正面2	-	-	-	-	o	-	-	-	-	-	-	-	-
見下ろし1	-	-	-	-	-	-	-	-	-	-	-	-	-
見下ろし2	-	-	-	-	-	-	-	-	-	-	-	-	-
様々な角度	-	-	-	-	-	-	-	-	-	-	-	-	-

表 4.2: 識別時の検出結果

個体番号	00	01	02	03	04	05	06	07	08	09	10	11	12	13
正面1	o	o	o	o	o	o	o	o	o	x	o	x	o	
正面2	-	-	-	-	-	-	-	-	-	x	-	x	-	
見下ろし1	-	-	-	-	-	-	-	-	-	o	-	o	-	
見下ろし2	-	-	-	-	-	-	-	-	-	-	-	-	-	
様々な角度	-	-	-	-	-	-	-	-	-	-	-	-	-	

個体番号	14	15	16	17	18	19	20	21	22	23	24	25	26
正面1	o	o	o	o	o	o	o	o	o	o	o	o	o
正面2	-	-	-	-	-	-	-	-	-	-	-	-	-
見下ろし1	-	-	-	-	-	-	-	-	-	-	-	-	-
見下ろし2	-	-	-	-	-	-	-	-	-	-	-	-	-
様々な角度	-	-	-	-	-	-	-	-	-	-	-	-	-



図 4.3: 正面と見下ろしによる撮影で検出できなかった R09 と R11

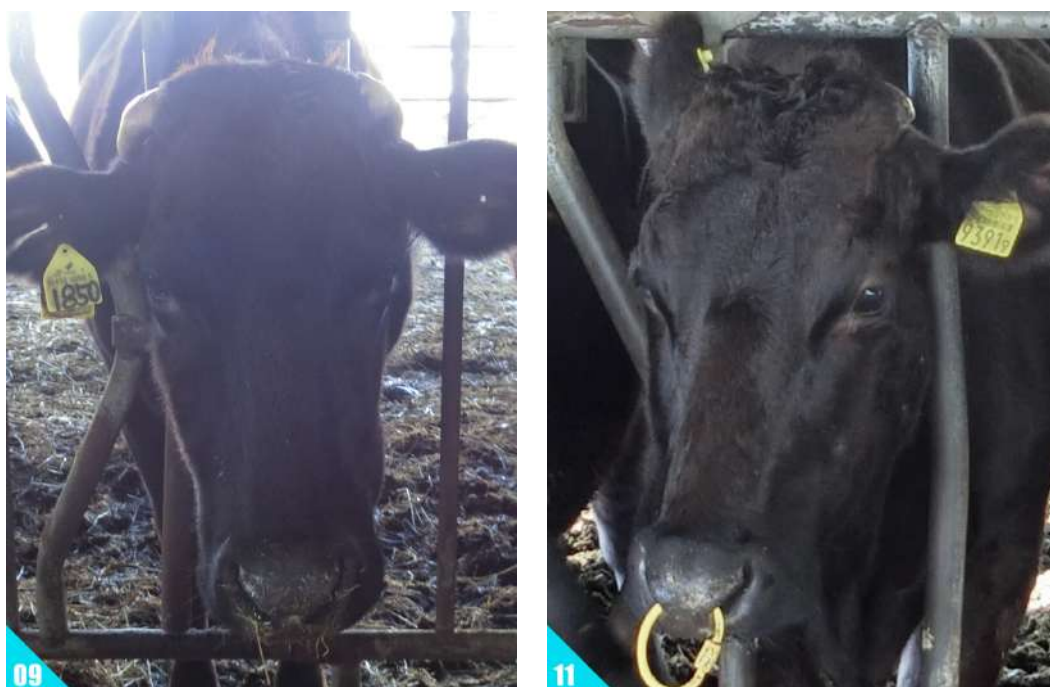


図 4.4: 見下ろしによる撮影で検出できた C09 と C11





図 4.5: 検出ができなかった R13

### 4.3 識別性能

識別性能は、R00～R26 を登録したのち、C00～C26 の識別が正しく行えるかを評価する。なお、R13 の牛顔検出ができなかったことにより、13 は識別性能の評価対象から除外した。

実験は、CowFindAR による撮影での実験に加えて、高宗の研究で行われた実験を模した、手作業により切り出された牛顔画像を識別器に直接与えた実験も行う。特徴ベクトルの抽出を行う上で入力画像の状態が最も良好である結果と比較して、CowFindAR での識別がどの程度劣化するかという観点で評価する。

結果を表 4.3 に示す。○と表記された項目は正しく識別できたことを、数字が書かれた項目は誤った識別を識別結果とともに示している。CowFindAR は 26 枚の画像のうち 14 枚 (53.8%) を、直接与えた場合は 26 枚のうち 16 枚 (61.5%) を正しく識別した。この結果から、画像の品質は識別精度に大きな影響は及ぼさないと考えられる。

また、識別に成功した個体の多くは共通して成功している。高宗の研究では、肉牛 1 頭につき 6 枚の画像を登録して 92.8% の精度を示していたため、CowFindAR においても

1 頭あたりの登録数を増やすことでより高い識別精度を期待できる。

表 4.3: 識別精度評価実験の結果

個体番号	00	01	02	03	04	05	06	07	08	09	10	11	12
CowFindAR	14	14	04	20	08	o	o	o	06	o	o	17	24
直接	o	o	18	02	o	o	o	o	06	o	06	23	o

個体番号	14	15	16	17	18	19	20	21	22	23	24	25	26
CowFindAR	o	o	o	o	02	o	o	05	o	o	o	19	14
直接	o	o	o	00	05	o	o	25	o	01	o	o	14

## 4.4 レスポンスタイム

サーバシステムを処理速度の観点で評価するために、レスポンスタイムの計測を行う。レスポンスタイムは識別機能利用時に対して計測する。

レスポンスタイムは要求を送ってから応答を受け取るまでの時間であり、クライアントシステムとサーバシステム間の通信にかかる時間も含むものである。クライアントシステムで図 4.6 に示すレスポンスタイム計測開始時点とレスポンスタイム計測終了時点での時刻を取得し、差分をとることで測定する。牛顔検出の成否によっては特徴ベクトルの抽出以降の処理が行われないこともあるため、牛顔検出に成功した場合のレスポンスタイムと牛顔検出に失敗した場合のレスポンスタイムは別の数値として測定する。

シナリオに則った測定をした場合、牛顔検出に成功した場合のレスポンスタイムは 26 回測定されるが、牛顔検出に失敗した場合のレスポンスタイムは回数が定まらない。今回の実験では牛顔の検出に 4 回失敗しているため、牛の映っていない画像をシナリオの撮影に追加して 22 回撮影する。

それぞれ 26 回ずつ測定したグラフを図 4.7 に示す。青色の線が牛顔検出に成功した場合、赤色の線が牛顔検出に失敗した場合で、実線が測定値、破線が平均値である。縦軸の単位はミリ秒である。牛顔検出に成功した場合の平均値は 279 ミリ秒、牛顔検出に失敗した場合の平均値は 183 ミリ秒であった。

牛顔検出の成否により実行の有無が変化する処理は特徴ベクトルの抽出と特徴ベクトル

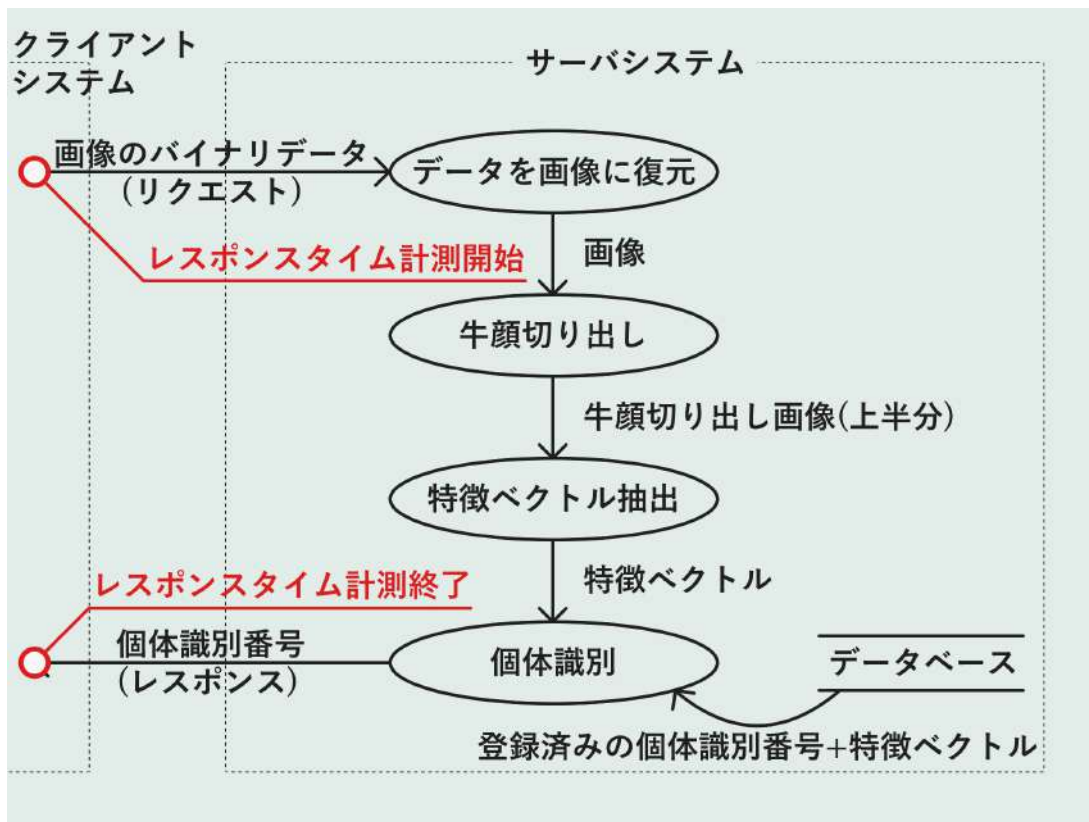


図 4.6: レスポンスタイム計測のタイミング

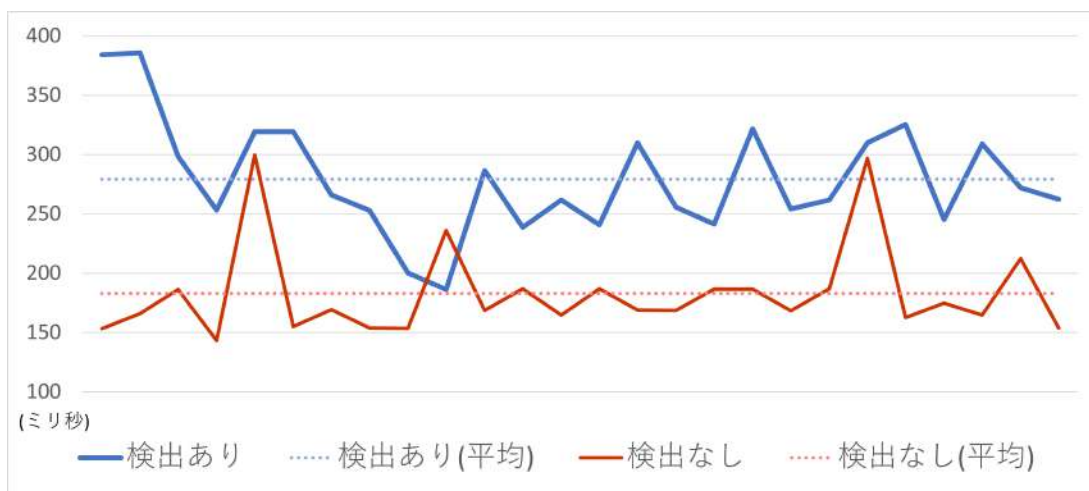


図 4.7: レスポンスタイム計測結果

の比較による識別であるため、平均レスポンスタイムの差分から、これらの処理時間はおおよそ 100 ミリ秒前後であると推定できる。特に特徴ベクトルの比較は登録されている牛顔数 × 4096 個の値を扱った処理であるため、登録数の増加によっては処理時間への影響が大きい可能性がある。

そこで、26 頭分の登録データをダミーとして 5 倍、10 倍、25 倍、50 倍に複製し、検出処理をそれぞれ 10 回ずつ行うことで、登録数の増加による処理時間への影響を測定する。それぞれ 10 回分の測定結果の平均を算出し、横軸をデータ枚数、縦軸をレスポンスタイムとしてプロットしたものを図 4.8 に示す。この結果から、登録データ枚数に比例してレスポンスタイムが増加することが分かった。

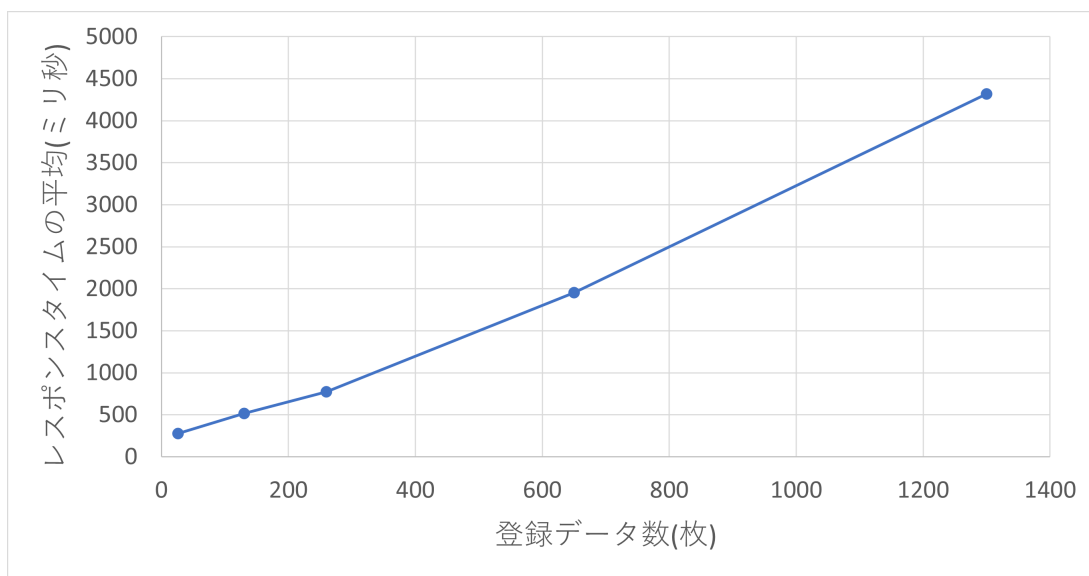


図 4.8: 登録データ数の増加に対するレスポンスタイムの推移

サーバシステムの処理時間をさらに分析するために、通信、バイナリデータからの画像復元処理、牛顔切り出し、牛個体識別にかかった処理時間を計測し、積み上げグラフとして図 4.9 に示す。この結果から、HTTP 通信に伴うバイナリデータからの画像の復元処理に最も時間がかかっている。また、個体識別にかかる時間のうち、ほとんどが SVM による特徴ベクトルの比較であるため、処理の高速化には他の比較手法を用いた実装も必要である。

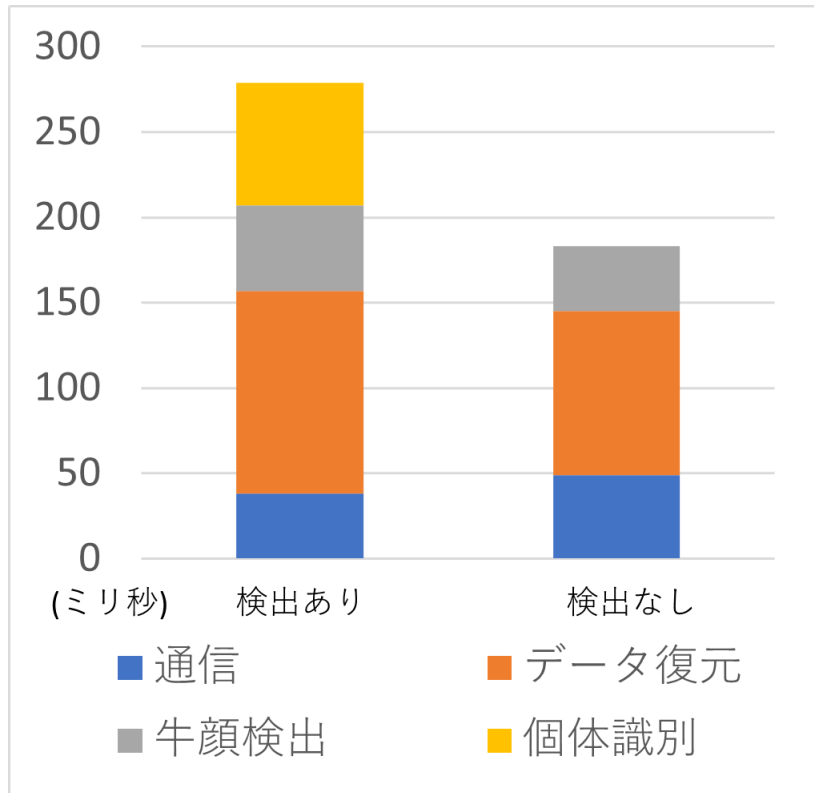


図 4.9: サーバシステムにおける処理ごとの実行時間

## 4.5 考察

これらの評価実験から、識別精度を上げるためには登録データ数を増やす必要があるが、登録データを増やすと処理にかかる時間が長くなってしまいうため、識別精度と処理速度がトレードオフの関係になっていることが分かった。

処理精度の面ではデータを増やすほかに、牛顔専用の特徴ベクトル抽出ネットワークを既存のネットワークからファインチューニングするという手法がある。

このサーバシステムは、牛顔切り出しをしている YOLOv5 のネットワークを他の畜産動物用に学習したものに変えることで、牛以外の畜産動物への利用も可能であるが、他の畜産動物では特徴ベクトル抽出器が捉える特徴も異なる可能性がある。

## 5 ARによる位置合わせの正確性の検討

### 5.1 実験方法

ARシステムにおいて情報の位置合わせは重要な課題である。この実験では、CowFindARにおけるARシステムであるシームレスモードについて、情報の位置合わせが正確に行えるか評価する。CowFindARでは、レスポンスのたびに検出結果を受け取って位置合わせを行うため、正しい位置に情報を配置できていない場合、リクエストからレスポンスまでの間に位置合わせにズレが発生する可能性がある。そこで、牛顔を映した状態でスマートフォンを動かすことにより、クライアントシステムが原因となる位置合わせのズレを評価する。

評価実験では、結果を正確に評価するために、システムに変更を加える。図5.1のような、個体識別番号を示すオブジェクトではなく、牛顔検出領域を示す半透明の赤色の矩形を表示する。また、レスポンスタイムが長い場合を想定するために、登録データ枚数を650枚にする。CowFindARのクライアントシステムはZenfone8に実装した。被写体として原寸大に近い牛の顔画像を用意するために、牛顔の縦幅が0.5mになるように縦向きの画面に牛顔画像を映す。



図 5.1: 表示の変更について

実験は、牛顔とスマートフォンの距離が  $1m$  の場合と  $1.5m$  の場合について、以下のシナリオに則って行う。

- ① 牛顔をスマートフォン画面の中心に映してしばらく待つ
- ② 牛顔をスマートフォン画面の縁に添って3周、牛顔に対してスマートフォンを平行移動する

1 番目の手順では静止した状態での位置合わせ精度、2 番目の手順では位置変化に対する位置合わせの精度が評価できる。

実験では牛顔とスマートフォンの距離を推定した提案手法のシステムと比較するために、距離を  $1m$  と仮定した場合でも実験を行う。

## 5.2 実験結果

実験の結果はスマートフォンの画面録画機能でシナリオを通過する様子を記録する。この動画を目視で確認し、静止した状態、牛顔画像が右辺、下辺、左辺、上辺のそれぞれに映っている場合において、最も結果が悪いと思われるフレームをそれぞれ抜き出した合計5フレームに関して比較評価を行う。

牛顔とスマートフォンの距離を仮定した場合において、実際の牛顔とスマートフォンの距離が  $1m$  の場合の実験結果を図 5.2、 $1.5m$  の場合の実験結果を図 5.3 に示す。牛顔とスマートフォンの距離を推定した場合において、実際の牛顔とスマートフォンの距離が  $1m$  の場合の実験結果を図 5.4、 $1.5m$  の場合の実験結果を図 5.5 に示す。

実験から、距離を仮定したシステムでは、仮定した距離と同じ  $1m$  の距離では大きなズレはなかったが、仮定した距離よりも離れた  $1.5m$  の距離では牛の顔に対して半分以上のズレが発生した。牛顔の位置に対して情報を配置する位置がスマートフォンの近くにあり、画面内での牛顔の移動に対して情報を配置する位置の移動が少ないためである。

距離を推定したシステムに関して、 $1m$  での実験では距離を仮定したシステムと同程度のズレであったため、距離の推定は正しくできていると考えられる。 $1.5m$  での実験では距離を仮定したシステムと比較してズレが少なくなっていたため、距離の推定は効果的であると考えられる。



図 5.2: 距離を仮定した場合の、実際の距離が 1m の場合の実験結果



図 5.3: 距離を仮定した場合の、実際の距離が 1.5m の場合の実験結果





図 5.4: 距離を推定した場合の、実際の距離が 1m の場合の実験結果

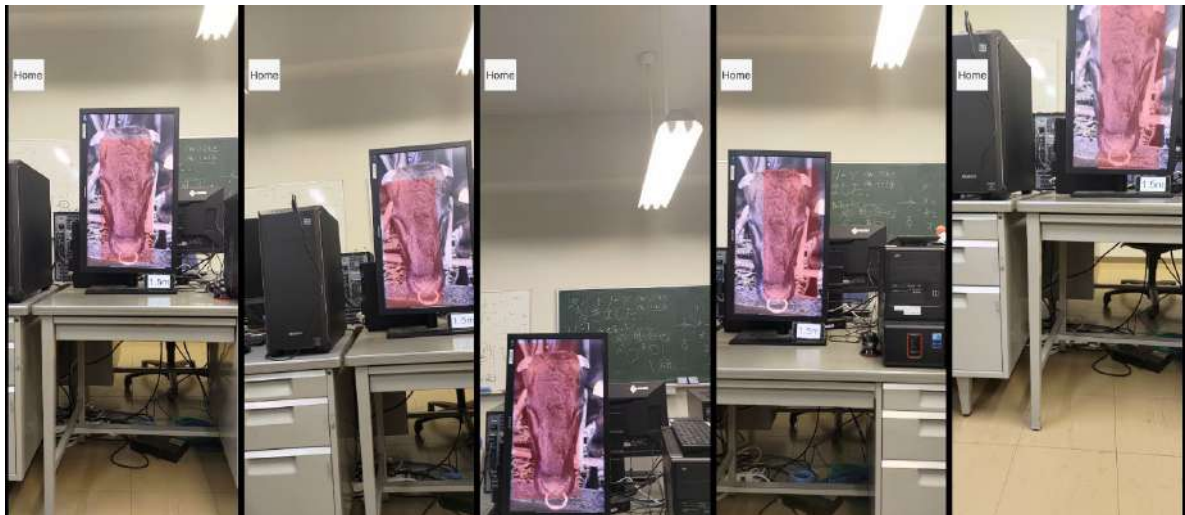


図 5.5: 距離を推定した場合の、実際の距離が 1.5m の場合の実験結果

### 5.3 距離推定の評価

牛顔とスマートフォンの距離を推定したシステムにおいて位置合わせを正確に行うためには、正確な距離推定が必要になる。そこで、実験中に推定した距離を記録・分析することで、距離推定の正確さについて評価する。推定した距離をグラフで表したものを図 5.6 に示す。距離の平均はそれぞれ、1m の実験では 1.008m、1.5m の実験では 1.517m だった。

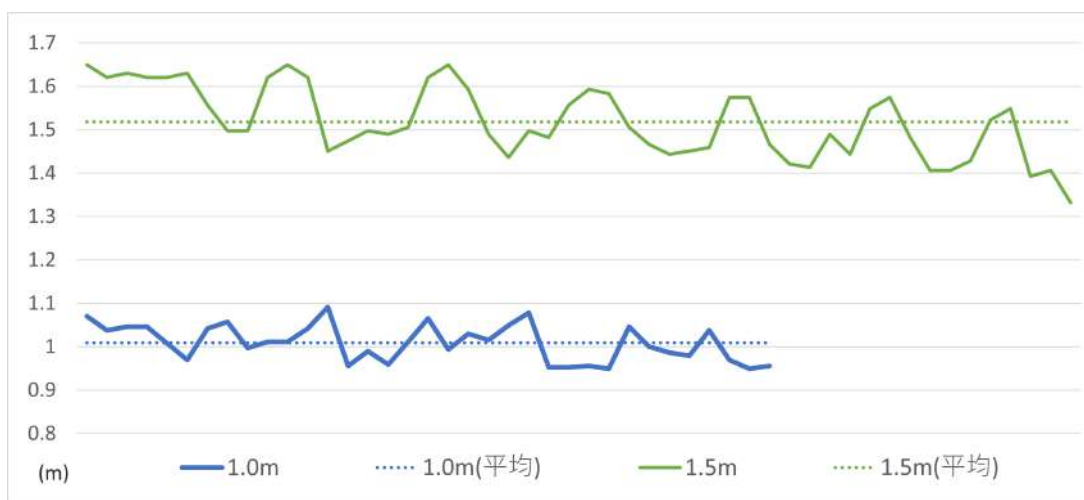


図 5.6: 距離推定のグラフ

この結果から、距離推定はおおむね正しく行えていることが分かった。特に 1.5m においては値の変動が激しいが、周期的に発生していることから、実験時に発生するスマートフォンの傾きによるものと考えられる。

## 5.4 考察

これらの実験の結果から、距離を仮定したシステムでは十分な位置合わせの精度にならないが、牛顔とスマートフォンの距離を推定することができれば、十分な位置合わせの精度が得られることが分かった。

しかし、実際の牛顔のサイズは牛個体によって変動するため、より高い位置合わせ精度を求める場合は、実際の牛顔の計測を大規模に行うことで、より信頼度の高い牛顔サイズの平均値を算出することや、牛個体の年齢や種類から牛顔サイズを推定する仕組みを作ることが必要だと考える。

## 6 おわりに

本論文では、牛顔個体識別を用いたモバイル端末向け管理情報提示システム「CowFindAR」を提案した。

提案手法は、スマートフォンで撮影した牛顔画像から、深層学習を用いて牛個体を識別し、ユーザに対する直感的な情報の提示に成功した。ステップモードは撮影した牛個体を登録し、正しいデータの登録ができたか確認できた。シームレスモードは識別した牛個体の情報を、AR システムとして正確な位置合わせ精度でユーザに提示できた。

サーバ性能の評価実験では、サーバで行う重要な処理である、牛顔切り出しと牛個体識別に関して、正しい出力を行う入力の傾向を分析した。また、いくつかの条件においてレスポンスタイムを計測することで、サーバの速度性能が変動する条件を調査した。評価実験の結果、識別精度と速度性能はトレードオフの関係になることが分かった。

シームレスモードの位置合わせ精度に関する評価実験では、距離を推定する提案手法と一定の距離を仮定する手法の比較を行った。実験では、提案手法がよりズレが少ない位置合わせを行えた。また、距離推定の値を分析し、提案手法が距離推定の手法として適切であることを示した。

今後の課題としては、識別精度の向上を目的とした牛専用の特徴ベクトル抽出ネットワークへの開発や、牛顔とスマートフォンの距離推定精度の向上、データベースへの登録件数が増えた際の識別処理の高速化、CowFindAR の畜産農業施設での運用などが挙げられる。

## 謝辞

本研究を行うにあたり、ご指導・ご協力いただいた椋木雅之教授、高塚佳代子技術職員、関口敏准教授に深く感謝申し上げます。

指導教員である椋木雅之教授には、学部から引き続き、研究に関する相談やアドバイスなど、丁寧かつ熱心にご指導いただきました。高塚佳代子技術職員には、ゼミや研究室内発表の際に沢山の助言をいただきました。関口敏准教授には、農学の側面から問題提起のヒントやデータの収集にご協力いただきました。

ご指導・ご協力いただいた皆様や、共に切磋琢磨した椋木研究室のメンバーの皆様に深く感謝を申し上げるとともに、皆様のご健勝とますますのご活躍をお祈り申し上げます。

## 参考文献

- [1] 農林水産省, “スマート農業”, <https://www.maff.go.jp/j/heya/sodan/17009/02.html>
- [2] cntxts Inc., “Smart Cattle<sup>®</sup>”, <https://smartcattle.net/home/index>
- [3] 株式会社ファームノート, “クラウド牛群管理システム「Farmnote Cloud」”, <https://farmnote.jp/cloud/>
- [4] 農林水産省, “牛・牛肉のトレーサビリティ”, <https://www.maff.go.jp/j/syouan/tikusui/trace/>
- [5] 独立行政法人家畜改良センター, “牛の個体識別情報検索サービス”, <https://www.id.nlbc.go.jp/top.html?pc>
- [6] Janina RUDOWICZ-NAWROCKA, Kornelia KUDLIŃSKA, Gniewko NIEDBAŁA, Magdalena PIEKUTOWSKA, “APPLICATION OF AUGMENTED REALITY IN DAIRY CATTLE MONITORING”, Journal of Research and Applications in Agricultural Engineering 2018, vol.63, pp.181-183
- [7] Thi Thi Zin, Moe Zet Pwint, Pann Thinzar Seint, Shin Thant, Shuhei Misawa, Kosuke Sumi, Kyohiro Yoshida, “Automatic Cow Location Tracking System Using Ear Tag Visual Analysis”, Sensors 2020, 20(12), 3564
- [8] 高宗 伸幸, “牛の正面顔画像による個体識別における転移学習の有効性評価”, 宮崎大学工学部情報システム工学科 卒業論文, 2020
- [9] IKEA, “IKEA Place”, <https://apps.apple.com/jp/app/ikea-place/id1279244498>
- [10] ナイアンティック, 株式会社ポケモン, “ポケモン GO”, <https://www.pokemongo.jp/>
- [11] Glenn Jocher, “YOLOv5”, <https://github.com/ultralytics/yolov5>
- [12] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, arXiv:1506.02640, 2015
- [13] 兒玉 光平, “一般物体認識による AR を用いた牛に対する情報の重畳表示”, 宮崎大学工学部情報システム工学科 卒業論文, 2020
- [14] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv:1409.1556, 2014
- [15] Unity Technologies, “AR Foundation”, <https://unity.com/ja/unity/>

features/arfoundation

- [16] ASUS, “Zenfone8”, <https://www.asus.com/jp/Mobile/Phones/ZenFone/Zenfone-8/>