

ドローンを使った牡蠣イカダメンテナンスのための 画像認識と飛行制御

Image Recognition and Flight Control for Oyster Raft Maintenance Using a Drone

家頭 裕也[†] 吉川 祐樹[†] 城明 舜磨[‡]

Yuya Yagashira Yuki Yoshikawa Shunma Jomyou

1. はじめに

小型無人航空機(以下、ドローンと呼ぶ)は空の産業革命と呼ばれ、ドローンが注目され始めた当初はホビー用途が多かったが、2016年頃から産業用ドローンの取り組みが世界的に加速している[1]。このような流れの中、我々の研究室では広島県呉市の牡蠣業者と連携し、ドローンを活用した牡蠣漁の業務サポートについて研究している。本稿では、牡蠣イカダのメンテナンス業務について、ドローンを活用することで牡蠣業者の業務負担を軽減する方法について報告する。

牡蠣を養殖している業者は、図1に示すように海上数キロの範囲に浮かぶ牡蠣イカダを船で回り、養殖作業を行っている。それに加えて、イカダの定期的なメンテナンスも行っており、特に大雨や高波、台風など悪天候の後には、牡蠣イカダの破損や漂流がないか確認する必要がある。このメンテナンスには、1つ1つイカダを見て回る時間と労力や船の燃料代のコストがかかるだけでなく、悪天候が続く場合には危険も伴う。このような牡蠣業者の課題について、ドローンが自律飛行しながら各イカダの状態を自動で空撮できれば、安全性の向上および時間と費用の削減に繋がると我々は考えた。

本稿では数値解析や行列計算で知られる MATLAB を用いてイカダの画像認識を行い、MATLAB のアドオンである Simulink を用いてドローンの飛行制御アルゴリズムについてシミュレーション上で検証を行う。

2. モチベーション

2.1 牡蠣イカダのメンテナンスと課題

広島県呉市音戸町の波多見湾では、図1に示すように牡蠣イカダが海上数キロの範囲に配置されている。広島県は牡蠣の養殖が盛んであり、県内の海上にはこのような牡蠣イカダの集合が多数存在する。図2はイカダ単体の写真で、大きさは一辺が10mから20m程度の四角形で、それぞれのイカダは互いにワイヤーで繋がっており両端のイカダは海底の重りと繋いで固定されている。音戸町には複数の牡蠣業者があり、各業者がイカダを列単位で管理しており、全体は1つの漁連や組合という形態となっている。

イカダの足場は竹で作られており、老朽化や腐食によって破損が起こる。またイカダ同士を繋ぐワイヤーは数年に一度交換するが、悪天候の影響で負荷がかかり切れることもある。牡蠣業者はこういったトラブルを未然に防ぐために定期的に自社が管理するイカダのメンテナンスを行っている。そのため、自宅や職場から漁場までの陸の移動時間、船での移動時間、船の燃料代など多くのコストがかかる。



図1 波多見湾に浮かぶ牡蠣イカダ



図2 牡蠣イカダ単体の写真

2.2 ドローンを使ったメンテナンスサポート

先に述べたメンテナンスの問題を解決するために、我々の研究室ではドローンを用いた牡蠣イカダのメンテナンスサポートについて研究を行っている。現在考えているメンテナンスサポートでは、ドローンが自律飛行しながら湾内に浮かぶ牡蠣イカダを自動撮影し、帰還後に撮影した写真をサーバーへアップロードする。これにより牡蠣業者はスマートフォンやタブレットを使って WEB 上で自社が管理するイカダの状態をいつでも確認することができる。このメンテナンスサポートを実現することで、船を出すことなくイカダの状態を把握できるため、移動時間や船の燃料の節約になる。また破損が確認できた場合には、破損状況に応じて修理に必要な資材を前もって準備することができ、破損したイカダを適切に修理することができる。

ドローンの空撮方法の概要を図3に示す。イカダは列ごとに並んでいることから、ドローンは基本前方に飛行する。イカダがカメラに写ったら左右の調整を行ってイカダの正面へ移動し、イカダの中心付近で止まって画像を撮影する。これを繰り返しながら各イカダを撮影していく。前方にイカダがなければ方向転換を行う。

[†] 呉工業高等専門学校

[‡] 広島大学

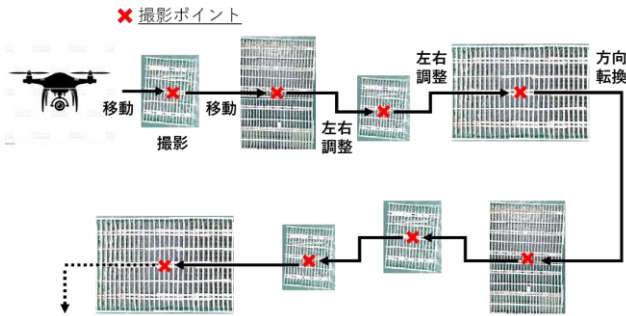


図3 ドローンを使ったイカダ撮影の概要

イカダは各列の両端のイカダが海底に繋がっているため大きく位置が変わることはない。そこで GPS 情報を使ってドローンの飛行経路を指定することで、イカダの各列の上空を飛行させることはできる。ただし、潮の流れる方向によって位置がずれるため、GPS を使った経路指定と撮影ポイント指定では正確にイカダの中心を撮影できる保証はない。そこでドローンのカメラで飛行画像を撮影しながら、その撮影画像に対してディープラーニングによるイカダの画像認識を行い、飛行経路の修正を行いながら各イカダの中心で撮影を行うよう制御する。飛行制御の方法については4章で述べる。

このメンテナンスサポートを実現するためには、以下の3つの課題がある。

- 牡蠣イカダの画像認識のためのディープラーニング
- ドローンの飛行制御アルゴリズム
- 撮影画像を提供する WEB アプリケーション

本稿では、A の課題を MATLAB で行う。課題 B は、A の画像認識を使った飛行制御アルゴリズムを提案し、Simulink を使ってシミュレーション上で検証する。実機を使った海上での飛行、および課題 C については今後の課題とする。

3. MATLAB を使った牡蠣イカダの画像認識

MATLAB [2] は数値解析や行列計算で知られるソフトウェアであり、本章では MATLAB を使ったディープラーニングによる牡蠣イカダの画像認識について述べる。

3.1 転移学習

本研究では、MATLAB のアドオンとして存在する AlexNet [3] と呼ばれる畳み込みニューラルネットワーク (CNN) をベースとして用いる。CNN は画像認識や音声認識によく利用されており、畳み込み層とプーリング層という層を用いるのが特徴である。AlexNet の場合、100 万枚を超える画像を学習させており(事前学習済みのネットワーク)、入力された画像を最終的に 1000 個のカテゴリに分類することができる。この AlexNet に対して牡蠣イカダと海の画像データを学習させることで、少ないデータ量でも高い精度で画像識別することができる。

次節では AlexNet をベースに転移学習を行うことで、ドローンの飛行画像に対して海と牡蠣イカダの判別を行う方法を述べる。

3.2 転移学習による牡蠣イカダの画像識別

まず始めに、入力となる牡蠣イカダの画像と海の画像をそれぞれ約 250 枚用意した。そして用意した牡蠣イカダと海の画像をそれぞれ学習用と検証用に分ける。画像の 70% を学習用に、残り 30% を検証用にそれぞれランダムに分け入力画像の準備を行う。

続いて牡蠣イカダと海を判別する転移学習について説明する。AlexNet (既存の学習済みネットワーク) は、最後の 3 つの層において 1000 個のカテゴリに分類するように構築されている。そこで今回は AlexNet を牡蠣イカダと海の計 2 種類のカテゴリに分類するよう再構築する。これにより、再構築した AlexNet に入力画像を与えた場合、その画像が牡蠣イカダなのか海なのか分類して出力する。

次に学習させる入力画像の設定を行う。入力層に入る画像のサイズを $227 \times 227 \times 3$ にする必要がある(3 は R,G,B)。更に入力画像に対して左右方向のランダムな反転や、入力画像に適用される部分を水平方向および垂直方向にランダムに平行移動させるといった処理を施す。これによって、学習の質が向上する。

最後に全体の学習における設定を行う。ミニバッチサイズ、エポック数、学習率を指定する。それぞれについて簡単に説明すると、ミニバッチサイズとは全画像データをいくつかのサブセットに分けた時のデータ数を指し、ミニバッチサイズを全体の近似として学習する。エポック数とは、入力画像をすべて学習した際にカウントする回数のごとで、1 エポックだと入力画像を全て学習したのは 1 回だということを表す。学習率とは、学習を行うごとに層の中での特徴抽出のためのパラメータを調整する値のごとである。本研究では、ミニバッチサイズを 10、エポック数を 6、初期学習率を e^{-4} (0.018) とする。

転移学習を用いて、入力画像の学習と検証を行った。画像が牡蠣イカダと海の 2 種類であるため、学習が初期の段階での検証精度が非常に高く、損失に関しても 0 に近い値を出した。設定ではエポック数を 6 にしていたが、1 エポック内で学習が完結したため手動で学習を止めた。これは学習に偏りが生じることを防ぐためである。結果として、学習の経過時間は 44 秒で検証精度は 100% となった。

3.3 画像認識にかかる時間と精度の関係

図 4 は、牡蠣イカダが写っているある画像 (大きさは 3840×2160 ピクセル) について、前節で構築した転移学習を用いてその画像の中でイカダと判別された箇所を赤く着色した結果である。この処理は、まず転移学習によってイカダか海かを判別する部分画像の単位 (例えば $400\text{pix} \times 400\text{pix}$ など) を決め、イカダが写っている画像について左上から右下にかけて列ごとに走査し、各部分画像がイカダか海かを判別する。もしその部分画像がイカダと判別された場合には赤く色を付ける。この転移学習によってイカダか海かを判別する部分画像のことをこれ以降はスキャン画像と呼ぶ。

上記の処理に関して、スキャン画像のサイズが大きいかほど画像全体に対する転移学習の回数は減るため計算時間は短くなるが、その一方でイカダと海の境界はイカダと認識される可能性があるため精度は粗くなる。逆にスキャン画像のサイズが小さいほど画像全体に対する転移学習の回数

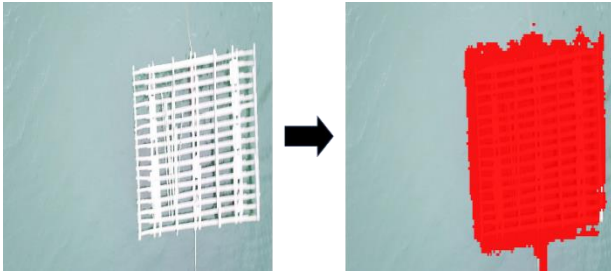


図 4 転移学習によるイカダの位置認識の例

表 1 イカダの位置認識にかかる時間の結果

判別結果	スキャン画像サイズ	オーバーラップ率	時間
A	400×400	50%	2.8秒
B		25%	1.5秒
C		0%	1秒
D	500×500	50%	2秒
E		25%	0.8秒
F		0%	0.6秒
G	800×800	50%	0.7秒
H		25%	0.5秒
I		0%	0.4秒

は増えるため計算時間は長くなるが、その一方で精度は細くなる。また、走査させるときに前のスキャン画像と次のスキャン画像を重ねることによって、時間と精度は変わる。このスキャン画像の重なる比率をオーバーラップ率と呼ぶ。オーバーラップ率が高いほど転移学習の回数が増えるため、時間は長くなるが精度は細かい。一方でオーバーラップ率が低いほど転移学習の回数が減るため、時間は短くなるが精度は粗い。

以下に、イカダの位置を認識する処理について実験結果を示す。表 1 は、スキャン画像サイズとオーバーラップ率を変化させた場合の計算時間の結果を示している。また図 5 はスキャン画像サイズとオーバーラップ率を変化させた場合のイカダの位置認識の結果を示している。この実験では、スキャン画像サイズは 400×400, 500×500, 800×800 (単位はピクセル) の 3 パターンとし、オーバーラップ率は 0% (重なりなし)、25% (重なり 1/4)、50% (重なり 1/2) の 3 パターンとした。

例えば同じスキャン画像サイズ 400×400 の A, B, C の結果を比べると、オーバーラップ率 50% で計算時間は 2.8 秒、25% で 1.5 秒、0% で 1 秒という結果だった。一方で、スキャン画像サイズ 800×800 にすると、オーバーラップ率 50% で計算時間は 0.7 秒、25% で 0.5 秒、0% で 0.4 秒という結果だった。続いて図 5 のイカダの位置認識の結果について比較する。スキャン画像サイズ 400×400 の A, B, C (オーバーラップ率は順に 50%, 25%, 0%) とスキャン画像サイズ 800×800 の G, H, I (オーバーラップ率は順に 50%, 25%, 0%) を比べると、同じオーバーラップ率では 400×400 の方がイカダの輪郭に沿ってきめ細やかに識別できている。一方 800×800 はイカダの輪郭よりムダに大きく認識している。またオーバーラップ率については、必ずしも高い方が正確に認識しているとは言えない結果となった。

この結果から、4 章以降の飛行制御シミュレーションでは、画像サイズ 800×800、オーバーラップ率は 0% で進めることとする。

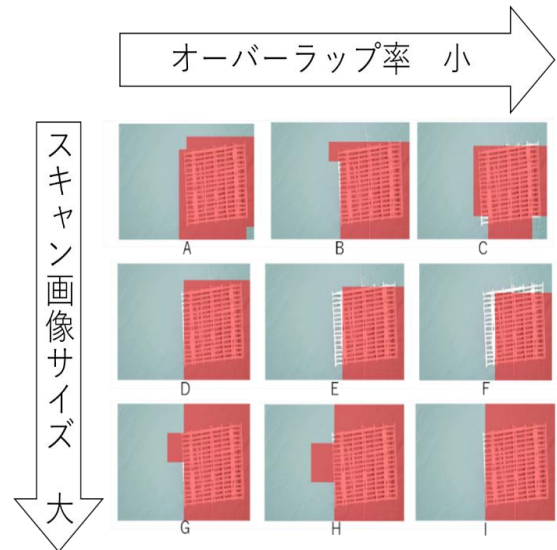


図 5 イカダの位置認識の結果

4. Simulink を用いたドローンの飛行制御

Simulink とは、モデリング、そしてシミュレーションを行うことができる MATLAB のアドオンである。Simulink では、様々な機能を持ったブロックを組み合わせることでシミュレーションを行うことができ、更に作成したブロック線図は C 言語に変換することもできる。またブロック線図の動作フローは Simulink にサポートされているハードウェアに実装することができる。本研究では、Simulink でサポートされているドローン (Parrot Minidrone MAMBO FLY [4]) を使用する。シミュレーションの環境は、海の画像の上に牡蠣イカダの画像を配置することで Simulink 上に仮想空間を作り、Parrot Minidrone に搭載されている下向きカメラで牡蠣イカダを順に撮影しながら飛行する制御アルゴリズムを考える。

Simulink のシミュレーションにおいて Parrot Minidrone のカメラ画像に対して画像認識を行うには MATLAB と連携する必要がある。MATLAB では、カメラから取得した動画 (連続的な画像) を転移学習にかけることでその画像に牡蠣イカダが写っているかどうか判別し、その結果を Simulink に出力する。その連携を行うために図 6 に示すように、Simulink 上に転移学習の結果を渡す機能を持つブロックを作成した。そのブロックを <MATLAB System> とし、その中に <alexTest.m> というプログラムファイルが動作する形となっている。

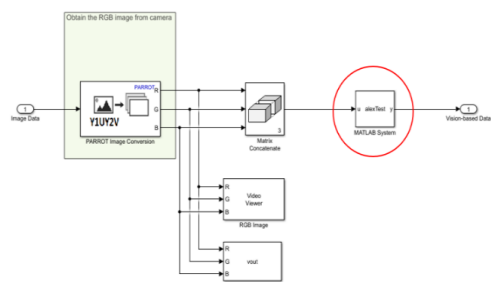


図 6 Simulink 内の MATLAB System

またシミュレーション環境は以下を前提とする。実際の牡蠣イカダは図 1 に示すように海に配置されているが、本シミュレーションではまず一列に並んだイカダのみを対象として飛行制御を考える。そこで Simulink の VR 空間にイカダの画像を一列に並べて配置し、背景は波多見湾の海画像とする。一列に並んだイカダはおよそ直線上にあるが、各イカダは潮の影響を受けて位置がずれることを想定し、ドローンの進行方向に向かって左右にずらして配置する。各イカダは実際に撮影した複数の画像を使用し、形は正方形または長方形のものがある。

4.1 飛行制御アルゴリズム

本節では異なる 2 つの飛行制御アルゴリズムを提案する。飛行制御アルゴリズムの設計は Stateflow によって行う。Stateflow とは、Simulink モデル内のブロックとしてシミュレーションができる、または MATLAB のオブジェクトとして実行できるといった用途に用いられるツールである。Stateflow では、状態遷移図やフローチャートといったグラフィカル言語を提供する。Simulink と同様ブロックを組み合わせてシミュレーションを行うが、Simulink と違い、自身でブロックを作成、変数設定などを行う必要がある。本研究では、Stateflow を Simulink モデル内のブロックとして使用し、MATLAB との連携によって ParrotMinidrone の飛行制御を行う。

4.1.1 カメラの画像を 4 分割し画像認識を行う飛行制御

Simulink によって取得したカメラの映像(画像の集まり)を MATLAB(alexTest.m)によって 4 分割し、それぞれに対して転移学習させる。取得した画像サイズは 120×160 であり、4 分割の場合、1 つあたりの画像サイズは 60×80 である。画像を左上、右上、左下、右下と分割することができ、それぞれに牡蠣イカダが写っているかを判別させて、その結果としてある一定の値を出力する。値を出力するためには 5 つの条件を満たす必要がある。画像の分割・画像認識(転移学習)・条件設定・値の出力すべてを MATLAB のプログラムである alexTest.m で行う。

Simulink ではドローンを離陸させた状態から一定の速度で真っ直ぐに直進することを軸とするプログラムを作成する。ドローンの飛行中カメラが牡蠣イカダ全体を収められていない場合、微調整を行う必要がある。表 2 より、出力された値を Stateflow に送ることで左右への移動や、空中で停止といった命令を行う。例えば 2 つ目の「左上と左下を牡蠣イカダと認識した場合」のとき、1 という値を Stateflow に出力し、ドローンに左へ移動する命令を出す。3 つ目についても、3 つ目と同様、2 という出力によって右へ移動させる。微調整であるため、速度は直進に比べて小さく設定する。1 つ目の条件を満たしたとき 100 という値を出力し、ドローンのカメラのシャッターを切るため空中停止を命令する。前提としてほぼ一直線上に牡蠣イカダが配置されているため、軌道修正は行わず、そのまま直進する。4 つ目の条件は、牡蠣イカダが何もない(カメラ全体に海が写っている)ときに主に出力される。シミュレーション時間は 200 秒に設定した。

表 2 alexTest.m で設定した条件と実行 (画像 4 分割)

	条件	実行
ケース1	左上, 右上, 左下, 右下 すべて牡蠣イカダと認識した場合	"100"を出力
ケース2	左上と左下を牡蠣イカダと 認識した場合	"1"を出力
ケース3	右上と右下を牡蠣イカダと認 識した場合	"2"を出力
ケース4	それら以外の場合	"0"を出力

4.1.2 カメラの画像を 16 分割し画像認識に加えて重み行列を使う飛行制御

ここでも alexTest.m を用いて転移学習による牡蠣イカダの画像認識と画像判別を行うが、前者の alexTest.m とは中身が異なる。まず、分割数を変更し、取得した画像を 16 分割する。そして重み行列を用いてカメラの情報を数値化する。重み行列とは、情報の大きさ・重要度を数値として表した<重み>を使った行列のことであり、この重み行列と 16 分割したカメラの情報(0 と 1 の 4×4 行列)の要素積が牡蠣イカダの位置を示した行列となる。本研究では空中停止や左右に移動してほしいときに用いる。重み行列の設定と、具体的な手法は以下に記す。

牡蠣イカダが図 7 のように写っていると仮定する。本研究で設定した重み行列について、上から中央、左上、右上に重みがある行列となっている。それらの行列に、16 分割された牡蠣イカダの判別結果の行列を要素同士で掛け合わせる。この要素積をそれぞれ M 行列, L 行列, R 行列と置く。また M, L, R 行列の各要素の合計を M, L, R という変数に代入し、この値を条件に用いる。図 7 の場合は $M=60$, $L=1$, $R=2$ である。

表 3 に条件と実行を示す。n という変数は M 行列の重みのある範囲に写っている牡蠣イカダ(値は 1)の要素の和である。図 7 でいうと $n=4$ である。M/n は、牡蠣イカダが重みのある範囲に写るほど値として小さくなる。牡蠣イカダが中央に写るほど M が大きくなり、値としても大きくなる。図 7 では $M/n=60/15=4$ である。また牡蠣イカダが写っていない場合、 $n=0$ となり M/n は数学的に間違っている。そのため予め n に 0.0001 という小さい値を加えておく。

Stateflow による飛行制御については、前者の手法と同じく真っ直ぐに直進することを軸とする。表 3 の 1 つ目の条件は右上に重み、つまり画像の右上に牡蠣筏が存在していることを意味している。そのとき 1 という値が Stateflow に送られて、右に移動することを命令する。2 つ目の条件も同様、左に移動することを命令する。3 つ目の条件では、 $R-L=0$ より牡蠣筏が中央 2 列の範囲に位置しており、且つ牡蠣筏が真ん中に写っていないということを表す。このときはそのまま直進することを命令する。4 つ目は 3 つ目の逆で、牡蠣筏が真ん中に写っているときを表しており、空中停止を命令する。また注意点として $R-L=0$ は牡蠣筏が写っていないときでも満たす。まとめると、直進することを軸にカメラによって取得した画像や表 3 をもとに左右の移動や空中停止の命令を行う。

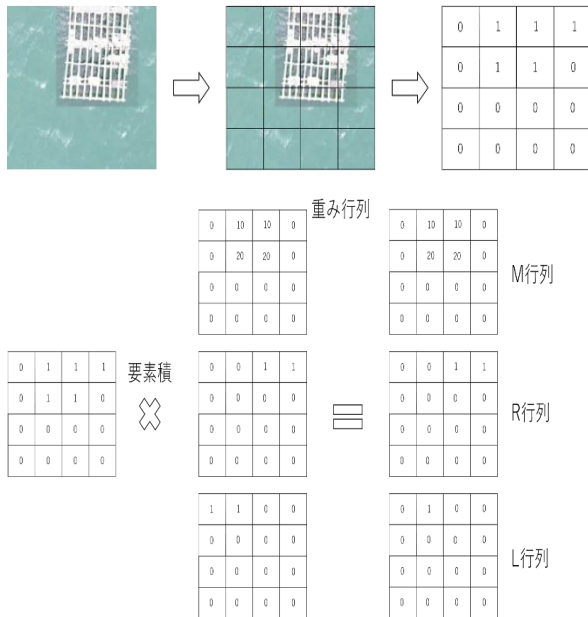


図 7 重み行列を用いた M,L,R 行列の作成

5. 結果と考察

2つのシミュレーションについて結果と考察を述べる。

まず 4.1.1 節で述べたカメラの画像を 4 分割し画像認識を行う飛行制御では、カメラに写る牡蠣イカダの大きさによっては、想定通りの動きにならないことがあった。小さいイカダの場合、例えばドローンのカメラの右側にイカダが写ったら右へ移動するが、フィードバック制御ではないため右へ行き過ぎることがあった。すると今度はカメラの左にイカダが写り左へ移動するよう命令される。この繰り返しによってドローンがループに陥ることがあった。また他にも問題はあった。ドローンがあるイカダを撮影し次のイカダに移動する際、撮影済みのイカダと次のイカダが 2 つ同時にカメラに写った時に空中停止と撮影の動作が頻繁に起こった。これは我々が想定していなかった動作であり飛行時間のロスとなった。また、同じような画像を複数撮影するため、冗長な画像データが増えた。このような単純な制御はドローンの計算負荷が小さいという利点もある一方、目的を達成する飛行制御は難しいことが分かった。

次に 4.1.2 で述べたカメラの画像を 16 分割し、画像認識に加えて重み行列を用いた飛行制御では、ドローン前方の牡蠣イカダがカメラに写り始めると左右の位置を調整してイカダの中心に移動し、その後前進した。牡蠣イカダがカメラの中央に来ると空中停止し、撮影した後直進した。その後も同様に、前方にイカダが写ると左右の調整を行い前進、イカダ中央で空中停止と撮影を繰り返した。前者に比べてドローンの飛行制御のためにかかるコンピューターの負荷は大きいと予想するが、牡蠣イカダの真上を通る理想的な飛行を実現できた。

この飛行制御はカメラ画像を 16 分割しそれぞれのマスに重みを付けることで、イカダが画像のどの位置にあるかによって重みの総和が変わるようにしている。重みの総和に対して閾値を決めて飛行パターンを制御しているため、更に飛行制御の幅を広げることも考えられる。例えば、イカダが前方遠くに写っている時は飛行速度を上げて、カメラ

表 3 alexTest.m で設定した条件と実行 (画像 16 分割)

	条件	実行
ケース1	$R-L > 0$	"1"を出力
ケース2	$R-L < 0$	"2"を出力
ケース3	$R-L = 0$ かつ $0 < M/n < 15$	"3"を出力
ケース4	$R-L = 0$ かつ $16 < M/n$	"4"を出力
ケース5	それら以外	"0"を出力

中央に近づくにつれて速度を落とすことで、飛行時間を短縮できる可能性もある。更なる拡張は今後も課題として取り組んでいく。

6. まとめ

本研究では、転移学習による牡蠣イカダと海の画像認識を MATLAB で実装し、その画像認識を使ってドローンが海上のイカダの上を自動で飛びながら撮影する制御を提案した。Simulink での飛行シミュレーションでは、ドローンが想定どおりに各イカダの上空を飛行し撮影できることを確認できた。

ただし今回はシミュレーション上の飛行結果であり、実際の海上でドローンを飛ばすためには、風の影響や太陽光の海面反射、天候による輝度や色彩の違いなどリアルな環境に対応させる必要がある。また、現状ドローンの飛行時間は機体差があるものの、長くても 30 分程度である。そのためより短い時間で多くのイカダを撮影できるよう工夫する必要がある。これらを今後の課題とする。

謝辞

本研究について課題や貴重なご意見を頂き、またドローンの飛行試験のために漁場の飛行許可を頂いた音戸海産有限公司の栗原単様に深く感謝致します。

参考文献

- [1] 野波健蔵, “ドローン産業応用のすべて — 開発の基礎から活用の実際まで —”, オーム社, 2018 年.
- [2] MATLAB: <https://jp.mathworks.com/products/matlab.html>
- [3] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, NIPS, 2012.
- [4] Parrot MAMBO FLY: <https://hitecrod.co.jp/products/parrot/>.