

複数カメラを用いた野生動物警報装置の評価

佐藤 令音[†] 斎藤 寛[‡] 富岡 洋一[‡] 小平 行秀[‡]
 会津大学 コンピュータ理工学研究科[†]/コンピュータ理工学部[‡]

1. はじめに

近年、野生動物による人身事故や作物被害が問題となっている。この問題を減らすためには、野生動物の早期検出や対策が欠かせない。最近では、深層学習の発展により、カメラで撮影した画像に対して推論を行い、野生動物を自動で検出する手法が報告されている [1]-[3]。我々も [4]で深層学習を基にした野生動物検出システムを開発した。

[1]-[4]で用いられているトレイルカメラや検出装置は、正面に1つのモーションセンサーと1つのカメラを持ち、前方数 m から十数 m の範囲で野生動物を検出することができる。しかし、野生動物が動いていた場合、撮影した画像に写っていないことが起こり得る。

この問題を解決するために、本論文では、複数のカメラを用いて前方以外も検出することができる野生動物警報装置を提案する。図 1 (右)のように、モーションセンサーやカメラを3つ使用すれば、1つしか使用しない図 1 (左)と比べ、検出範囲を広げることができる。また、本論文では、提案する装置の消費エネルギーの評価結果を報告する。

2. 提案する複数カメラによる野生動物警報装置

本節では、提案する複数カメラによる野生動物警報装置の構成と処理フローを解説する。以下では、この装置のことを提案装置と呼ぶ。

2.1 構成

図 2 は、提案装置の構成を表す。提案装置は Raspberry Pi 3 Model B を中心としている。モーションセンサーは、装置のトリガーに使用する。照度センサーは、日中か夜間かの判断のために使用し、赤外線投光器は、夜間撮影に使用する。検出情報は、同一 Wi-Fi ネットワーク内にある 3G/LTE モジュールを搭載している親ノードを経由し、サーバーに送信する。回転灯とスピーカーは、検出時の音や光による警報のために用いる。警報装置の電源は、商用電源がないところでの使用を想定し、ソーラーパネルと鉛蓄電池、バッテリーコントローラーからなる。

2.2 処理フロー

本論文では、逐次処理とマルチプロセスによ

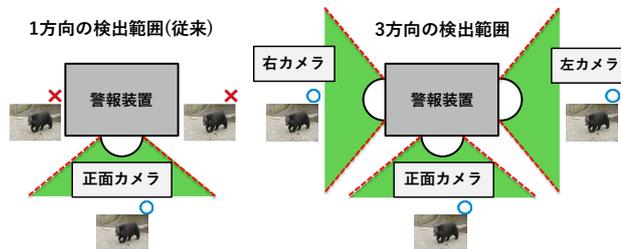


図 1 従来の装置と提案装置。

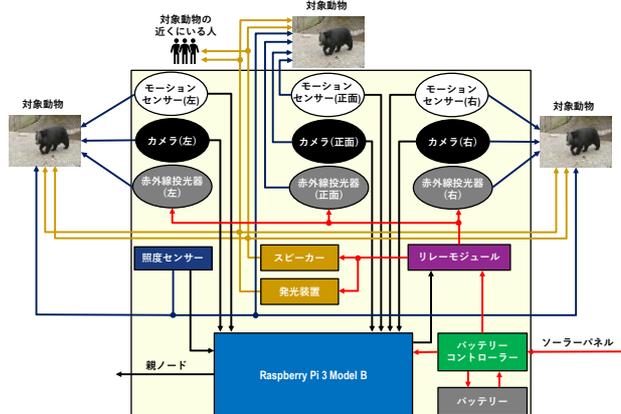


図 2 提案装置の構成図。

る並列処理の2つの処理フローを検討する。並列処理の方が高速化を期待できるかもしれないが、OS にて管理するため、プロセスの生成や制御に時間を要すことも考えられる。そのため、逐次処理も検討した。

2.2.1 逐次処理

図 3 は、3 方向の逐次処理のフローを表す。逐次処理では、方向毎に処理の順番を定める。図 3 の場合、左、正面、右としている。3 つのセンサーのうち1つでもセンシングした場合、野生動物が動いていることを想定して全てのカメラで撮影を複数回行う。夜間の場合は、撮影の前に赤外線投光器をオンにする。夜間かどうかの確認と赤外線投光器をオンにするのは、カメラ撮影の前の照度センサーの処理で行う。次に、方向毎に撮影した複数枚の画像に対して、フレーム差分法を実行し、画像内で動きを検出する。動きがある場合、推論を行う。推論にて対象となる野生動物を検出した場合、即時に警報と周知を行うために、他の方向で撮影した画像に対する推論をスキップし、発光装置による光の出力から停止までを行う。

2.2.2 マルチプロセスによる並列処理

Raspberry Pi 3 Model B で使用されている ARM Cortex-A53 プロセッサは、4 つのコアを有して

Evaluation of a Wild Animal Alert System Using Multiple Cameras
 Reon Sato[†], Hiroshi Saito[‡], Yoichi Tomioka[‡], Yukihide Kohira[‡]
 Graduate School of Computer Science and Engineering, The
 University of Aizu[†], Department of Computer Science and
 Engineering, The University of Aizu[‡]

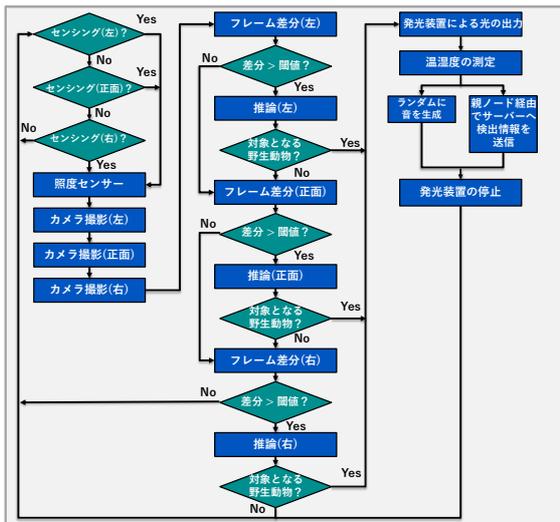


図3 逐次処理のフロー.

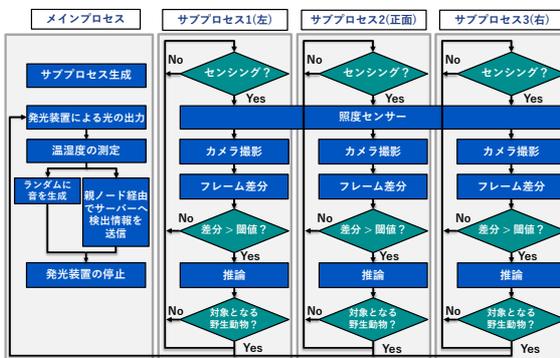


図4 並列処理のフロー.

いる. そのため, マルチプロセスにて各方向に対する処理を並列で実行する.

図4は, マルチプロセスによる処理フローを表す. メインプロセスでは, サブプロセスの生成と, 野生動物を検出した際の警報と周知を行う. サブプロセスは, 方向毎の処理を表す. どのコアを使うかは, Raspberry Pi OSが決める. なお, 逐次処理とは異なり, センシングした方向でしかカメラ撮影や推論は行わない.

3. 実験

実験では, 提案装置を逐次処理と並列処理で実行した際の実行時間, 消費電力, 消費エネルギーを評価した. また, カメラの個数による実行時間などへの影響を確認するために, 1方向のみ動作, 2方向動作, 3方向すべて動作させるようにした. 推論で使用した畳み込みニューラルネットワークモデルは, MobileNet-v2を我々の収集した野生動物画像を基に構成した画像データセットを用いて転移学習したものである. なお, あらかじめ準備したツキノワグマの写真をカメラとモーションセンサーの前にかざすことで提案装置を動作させている.

表1, 表2はそれぞれ, 逐次処理, 並列処理における実行時間 (T), 消費電力 (P), 消費エネルギー (E)を表す.

表1 逐次処理の実行時間, 消費電力, 消費エネルギー.

	T [s]	P [W]	E [J]
1方向のみ動作	2.7	2.3	6.2
2方向動作	4.8	2.5	12.0
3方向すべて動作	6.8	2.5	17.0

表2 並列処理の実行時間, 消費電力, 消費エネルギー.

	T [s]	P [W]	E [J]
1方向のみ動作	2.8	2.3	6.4
2方向動作	3.6	2.8	10.1
3方向すべて動作	4.3	3.3	14.2

実行時間は, センシングから推論が終了するまでの時間で, Python3で実装したプログラムに, perf_counter関数を挿入することで計測した. 消費電力の計測では, IndoorCorgiのESP-PowerMonitor [5]を使用し, 平均電流, 平均電圧から平均消費電力を求めた. 消費エネルギーは, 実行時間と消費電力の積である.

逐次処理, 並列処理共に, カメラの個数に比例して, 実行時間と消費電力が増加した. 一方, 2方向以上の場合, 消費電力は大きいが行時間が短い分, 並列処理は逐次処理と比べ消費エネルギーが少なかった. したがって, カメラを複数個用いるときは, 並列処理の方が良いということが言える.

4. おわりに

本論文では, 複数カメラを用いた野生動物警報装置を提案した. また, 提案装置の消費エネルギーを評価した. 実験より, 逐次処理より, マルチプロセスによる並列処理の方が消費エネルギーを削減できることを確認した.

参考文献

- [1] A. R. Elias et al., "Where's The Bear? - Automating Wildlife Image Processing Using IoT and Edge Cloud Systems", Proc. IEEE International Conference on Internet-of-Things Design and Implementation, pp.247-258, 2017.
- [2] N. Nonburinon et al., "A Novel Hierarchical Edge Computing Solution Based on Deep Learning for Distributed Image Recognition in IoT Systems", Proc. International Conference on Information Technology, pp.294-299, 2019.
- [3] B. H. Curtin and S. J. Matthews, "Deep Learning for Inexpensive Image Classification of Wildlife on the Raspberry Pi", Proc. IEEE Ubiquitous Computing, Electronics and Mobile Communication Conference, pp.82-87, 2019.
- [4] H. Saito et al., "Battery-powered Wild Animal Detection Nodes with Deep Learning", IEICE Transactions on Communications, Vol.E103-B, No.12, pp.1394-1402, Dec. 2020.
- [5] Indoor Corgi Elec., "E32-SolarCharger", <https://www.indoorcorgielec.com/products/e32-solarcharger/>