

要約

自然を相手にする農業従事者が抱える問題として獣害がある。獣害被害の中でも夜間に発生するものは、害獣の侵入に気づくことが困難であり、日中と異なり農業従事者が直接追い払うことが難しい。そのため基本的には獣害被害は事後に痕跡から経路や動物の種類を推定して電気柵や対策が取られている。しかし、害獣の正確な侵入経路がわからないため、柵の隙間といった不備のある位置がわからないことやそもそも推定した動物の種類が間違っているなど対策が機能しないことも少なくない。そのため適切な対策を取るために様々な対策を試す必要があり、大きな手間と時間・費用を要している。そこで我々はこれらの問題を解決するために、畑への侵入を検出・通知し、その動物の行動や大きさを推定することで、農業従事者の獣害対策を効果的に行えるよう支援するシステムを提案する。特に本論文ではクラウドサーバを用いて、畑への侵入したという情報および推定して得られた結果を保存するデータベース、畑への侵入情報から農業従事者へ通知する Web アプリケーション、推定された行動や大きさを地図を用いて行動履歴を表示する Web アプリケーションの設計を行った。機能としては、リバースプロキシ、WebPush を用いた通知サーバ、Leaflet.js による地図を用いた行動履歴表示を行う Web アプリケーション、データベースおよびデータベースアクセス制御用の内部 API があり、それぞれを Docker Compose を駆使して連携させたクラウドサーバを設計した。設計・実装後、クラウドサーバがやり取りする畑のゲートウェイに搭載される無線モジュールとの疎通確認および通信形式の確認、クラウドサーバ内の通知サーバからブラウザへの Push 通知の確認、クラウドサーバ内の Web アプリケーションによる行動履歴表示の確認を行った。さらなる実際の使用環境に合わせた改善をするには、全体を連携させて実際の畑で得られたセンサーデータで解析、通知・表示を行うことが必要である。展望としては、本論文担当であるクラウドサーバについてセキュリティや UI・UX の向上の他にリバースプロキシとデータベースアクセス制御用の内部 API が通信量のボトルネックになるといった懸念点が挙げられる。通信量のボトルネックは、本論文のシステム構成ではリバースプロキシと内部 API にデータが集中するのに対し 1 つずつしか用意していないことが根本的な原因である。そのため、kubernetes といった分散管理システムの導入することで、更なる冗長性とスケーラビリティが実現することができる。

参考文献

- [1] 農林水産省, “野生鳥獣による農作物被害状況の推移”, https://www.maff.go.jp/j/seisan/tyozyu/higai/hogai_zyoukyou/attach/pdf/index-31.pdf, 2024 年 1 月 18 日参照.
- [2] 大澤文孝, 浅井尚, “触って学ぶクラウドインフラ docker 基礎からのコンテナ構築”, 日経 BP マーケティング, 2020 年.
- [3] 掌田津耶乃, “Node.js 超入門”, 株式会社 秀和システム, 2017 年.
- [4] StrongLoop, IBM, “Express - Node.js web application framework”, <https://expressjs.com/>, 2024 年 1 月 18 日参照.
- [5] Volodymyr Agafonkin, “Leaflet - a JavaScript library for interactive maps”, <https://leafletjs.com>, 2024 年 1 月 18 日参照.
- [6] Mozilla Foundation, “プッシュ API - Web API | MDN”, https://developer.mozilla.org/ja/docs/Web/API/Push_API, 2024 年 1 月 18 日参照.
- [7] M. Thomson, E. Damaggio, B. Raymor, Ed., “Generic Event Delivery Using HTTP Push”, <https://datatracker.ietf.org/doc/html/rfc8030>, RFC8030, December 2016, 2024 年 1 月 21 日参照.
- [8] Mozilla Foundation, “サービスワーカー API - Web API | MDN”, https://developer.mozilla.org/ja/docs/Web/API/Service_Worker_API, 2024 年 1 月 18 日参照.
- [9] M. Thomson, P. Beverloo, “Voluntary Application Server Identification (VAPID)”, <https://datatracker.ietf.org/doc/html/rfc8292>, RFC8292, November 2017, 2024 年 1 月 21 日参照.
- [10] Nginx, “nginx”, <https://nginx.org/en/>, 2024 年 1 月 18 日参照.
- [11] Mozilla Foundation, “POST - HTTP | MDN”, <https://developer.mozilla.org/ja/docs/Web/HTTP/Methods/POST>, 2024 年 1 月 22 日参照.