

気象監視カメラを用いた災害監視システムの構築 — 深層学習による漏斗雲の自動抽出と追跡 —

藤井 祐貴[†] 本田 理恵[‡] 佐々 浩司[‡] 村田 健史[#] 村永 和哉[#] 有岡 無敵[§] 棚瀬 旺和[§]

[†] 高知大学総合人間自然科学研究科 〒780-8520 高知県高知市曙町 2-5-1

[‡] 高知大学自然科学系理工学部門 〒780-8520 高知県高知市曙町 2-5-1

情報通信研究機構 〒184-8795 東京都小金井市貫井北町 4-2-1

[§] 高知大学理学部応用理学科 〒780-8520 高知県高知市曙町 2-5-1

E-mail: [†] [‡] [§] {yukfujii, honda, b163k246, b163k121}@is.kochi-u.ac.jp [‡] sassa@kochi-u.ac.jp
{ken.murata, muranaga}@nict.go.jp

あらまし 近年、ネットワーク化された監視カメラ群によって取得された時系列画像から、気象災害につながる竜巻などの現象を検知しようとする試みが始まっている。我々はこのようなデータから竜巻の前兆となる漏斗雲を自動抽出するために、深層学習の1種である畳み込みニューラルネットワークによるオブジェクト抽出手法を検討している。本研究では階層化によるマルチスケールでのオブジェクト検出能力の向上、時間的隣接フレームの比較による時間連続性（同一性）の判定手法を検討するとともに、このプロセスを組み込んで、検出した漏斗雲のリアルタイム情報表示やアラートが可能な Web アプリケーションを構築した。このシステムを土佐湾沿岸に展開している実際の気象監視カメラ画像に適用した結果について報告する。

キーワード 漏斗雲, 竜巻, 気象監視システム, 深層学習, CNN, オブジェクト追跡

1. はじめに

近年ネットワーク化された監視カメラ群によって取得された時系列画像から気象災害につながるさまざまな現象を検知する試みが始まっている。このようなシステムにより広域かつ定常的な監視を行うことができるようになり、これまで見逃されていた事例を動画画像で記録することが可能となる。さらに取得されたリアルタイムのセンシングデータに機械学習の手法を適用することで災害の予兆を検知し、アラートに利用することが期待される。

高知大学では気象災害の中でも特に竜巻をターゲットとして監視カメラ群を竜巻発生頻度の高い土佐湾沿岸に24基展開して、常時観測を行っている。我々はこの監視カメラにより撮影された時系列画像から竜巻の特徴である漏斗状の雲(漏斗雲)を自動検出する手法を検討している。予備実験として深層学習によるオブジェクト検出手法であるYOLO(You only look once)^[1]を適用して異なるスケールの漏斗雲の検出を検討したが、精度再現率ともに課題が残り、また遠方の微小な漏斗雲の検出が難しいという問題があった^[3]。また隣り合う時間フレーム間での連続したオブジェクトの識別や追跡も実施できていなかった。

本研究では漏斗雲検出性能の向上のため、YOLOをあらかじめ階層化した画像に適用するマルチスケールでのオブジェクト検出と、新しいバージョンであるYOLOv3^[2]によるオブジェクト検出の2手法で性能評価した。さらに時間的隣接フレームの比較による時間連続性（同一性）の判定手法を検討した。また、このプロセスを利用して検出した漏斗雲のリアルタイム情

報表示が可能な Web アプリケーションを構築した。

2. システム概要

本研究では竜巻発生頻度の高い土佐湾周辺を監視するため、高知大学朝倉キャンパス、物部キャンパス、安芸、土佐清水、黒潮、須崎の6箇所にそれぞれ4方向に向けて配置されている監視カメラシステムの使用を想定する。これらの監視カメラの解像度はフルハイビジョンであり、撮像間隔は高知大学朝倉キャンパス、物部キャンパス、安芸の監視カメラについては1秒、土佐清水、黒潮、須崎に設置されている監視カメラについては1分である。図1に2017年9月11日に発生した竜巻を安芸の監視カメラで撮影された例を示す。ここでは雲底から突出した漏斗雲が観測点に近づき通り過ぎる様子が捉えられている^[4]。

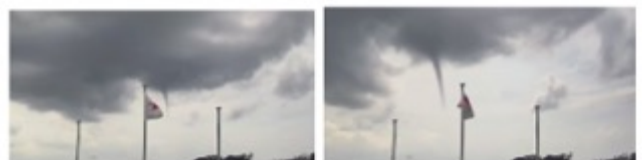


図1 気象監視カメラの画像例 2017年9月11日安芸 ch3(南東方向)で取得

図2に本研究で構築するシステムの概念図を示す。監視カメラからリアルタイムでネットワークを通じてサーバに画像が伝送される。サーバでは竜巻の前兆現象の自動抽出を行い、検出された解について同一性の判定を行う。そして検出された解を保存し Web アプリ

ケーションとしてユーザに検出結果を提示することを想定する。

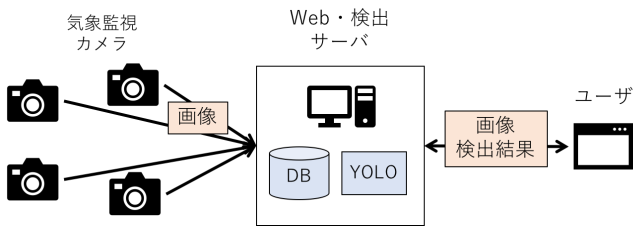


図 2 システムの概念図

3. 漏斗雲の時系列抽出

3.1. 漏斗雲検出

今回検出対象となる漏斗雲は外形が”漏斗状”という特徴を持つものの、高さや傾き、サイズ、色合いなどは事例によって異なる。このため、本研究では近年画像認識の分野で高速かつ高い検出性能を達成している深層学習の畳み込みニューラルネットワーク(CNN)によるオブジェクト検出手法である YOLO を採用した。YOLOには軽量なネットワーク構造をもつ Tiny モデルやマルチスケールに対応できるようにした YOLOv3 が存在する。

前述の通り、災害の予兆検出という観点では遠方で発生した微小なオブジェクト、漏斗雲から連続的に接近して通り過ぎる大きな対象まで連続的に抽出することが必要である。先行研究^[4]では遠方に出現した微小な対象を検出させるため YOLO の入力ユニットサイズ程度に画像をブロックとして分割したうえで、各ブロックに YOLO を適用していたが、逆に分割したブロックより大きい事例については検出ができない問題があった。そこで本研究においてはマルチスケールでの検出に対応するために以下に述べる 2 つの手法を検討した。

3.1.1. 階層的アプローチ

図 3 に本手法で認識時に用いる階層化とブロック化の概念図を示す。YOLO では画像に対して $S \times S$ のセルを設定し、セルが属するオブジェクトのクラスの識別率とセル毎のオブジェクトの領域と存在確率の組を B 個求めることができ、この結果を組み合わせることでオブジェクトの抽出と認識を同時に実行できる。また、本手法では固定サイズの画像を対象とした検出手法となっている。そのため、画像が CNN に入力される際には、画像サイズは入力ユニットのサイズに自動的にスケールされる。よって 入力ユニット数と画像サイズ、アスペクト比が大きく異なる場合、形状の特徴が失われてしまう。

この問題を解決するために、画像を入力ユニット相当のサイズに分割するものとする。その際、図 3 に示すように濃い青の部分が重複領域になるよう、点線で区切られた部分をブロックとして YOLO の入力画像としてサンプリングした。この際小さなものから大きな対象まで抽出できるように 2 階層の画像を準備した。サンプリングするには検出対象が見切れてしまう事例を回避するために外に 1 セル分余剰の領域を設定した。また、検出された事例は原画像の座標系上に統合した。

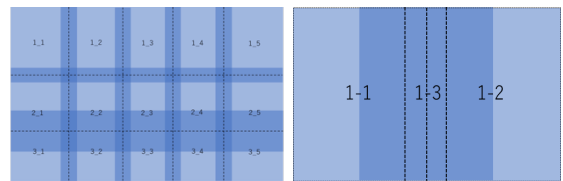


図 3 画像の階層化とブロック化。フルハイビジョンの画像におけるブロック化について示す。処理時には YOLO により全てのブロックを同じサイズにスケールされるため、左では小さい特徴を、右では大きい特徴が検出される。

図 4 には重複したセルにおける解の選定方法について示す。ブロック境界では解の棄却領域が互いに重なるように設定し、この領域に解の中心が存在する解は削除することで重複した解が削除できるようにした。これによりブロックの境界部でも空白領域が生じたり解が分裂したりすることなく、滑らかに解を求めることが可能になる。

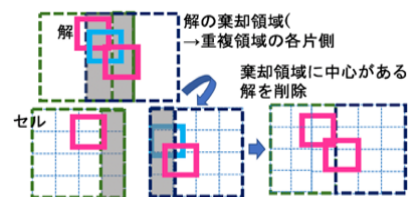


図 4 ブロック境界の重複部における解の棄却

さらに本研究では再現率を上げるために YOLO の検出しきい値を低く設定したが、そのことによって図 5 左のように多重解が発生してしまう場合があった。この問題は検出結果がオブジェクトの一部に反応することが多いため、図 5 右のように検出された解が重なり最小包含矩形(MBR: Minimum Bounding Rectangle)をまとめて 1 つの解とすることで対応した。



図5 初期の多重解(左)の MBR による整理(右)。マゼンタの矩形は抽出結果、緑の矩形は人間の視覚で漏斗雲と判定された領域。

3.1.2. 非階層的アプローチ

もう1つのアプローチとしてYOLOv3をそのまま用いて検出を行なった。YOLOv3ではYOLOをベースにネットワーク構造の変更や基準サイズとなるバウンディングボックスの導入などの工夫が追加された手法となっている。これらの工夫により入力する画像のサイズ、アスペクト比が異なる場合でも様々なスケール、形状の特徴を変更することなく検出できる。重複解については重複解の組の中で信頼度が低い結果を削除することで対応した。

なお、本研究ではフルハイビジョンの画像を入力した際GPUのメモリが不足して計算できなかったため、通常よりネットワークが軽量であるTinyモデルを使用した。

3.2. 漏斗雲の同一性判定

3.1.1 項、3.1.2 項で述べた検出手法により得られたフレームごとの解をもとにオブジェクトの同一性判定を行った。オブジェクトの同一性判定は隣接したフレーム間でオブジェクト位置がわずかにしか変化しないという特徴からIoU(intersection over union)を用いた判定を行なった。図6にその概念図を示す。現在のオブジェクト検出領域を $B(t)=\{B_i(t)|i=1,\dots,im\}$ 、前の時間に記録されたラベル付きオブジェクト候補のリストを $L(t-1)=\{L_j(t-1)|j=1,\dots,jm\}$ として、 $L(t-1)$ は{ID, 解の寿命 t_n , 最終時間の解領域 LB , 更新回数 t_s }として定義した。同一性判定では現在のオブジェクト検出領域と前の時間の追跡候補の領域リストとのIoUを計算し、しきい値 IoU_{th} 以上の解の組を同一オブジェクトと判定すると定義した。

$$IoU(B_i(t), LB_j(t-1)) = \frac{B_i(t) \cap LB_j(t-1)}{B_i(t) \cup LB_j(t-1)} \quad (1)$$

ここで解の寿命は追跡時にノイズなどにより一時的に追跡の途絶が発生したときも猶予を儲けるためのものである。(1)式により同一性が認められる解を検出した場合には解の寿命 t_n を解の寿命の最大値である T_n にセットし、座標を更新する。検出されない場合には解の寿命 t_n をデクリメントし、解の寿命 t_n が0となったも

のは次の時間のラベル付きオブジェクト候補リスト $L(t)$ に引き継がないことで解の追跡候補から削除した。また、検出された解が前の時間までの追跡候補との同一性が認められない場合は新規のラベルとして追加した。このとき、解の寿命は追跡断絶のしきい値である T_n を、解領域 LB は検出された領域を保存する。

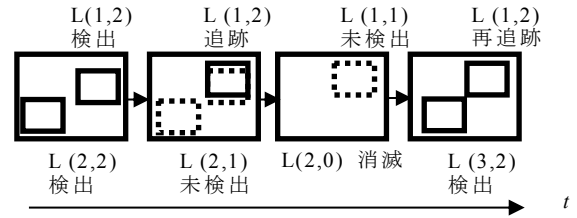


図6 解の寿命の最大値 $T_n=2$ における漏斗雲の追跡例。ラベル付きオブジェクト候補のリスト L の引数は、IDと解の寿命のみ記載

さらに、同一性判定結果を用いた誤検出解の削除についても行なった。図7に同一性判定結果を用いた誤検出解の削除法の概念図を示す。誤検出解の多くは図7左のように一時的に検出されるだけで、継続しないという特徴を持つ。そこで本研究では図7上のラベルID2のような、ラベル付きオブジェクト候補リストの更新回数 t_s が規定回数 T_s 未満のラベルの事例を削除することで誤検出解の削除を行うことで、図7下のよう

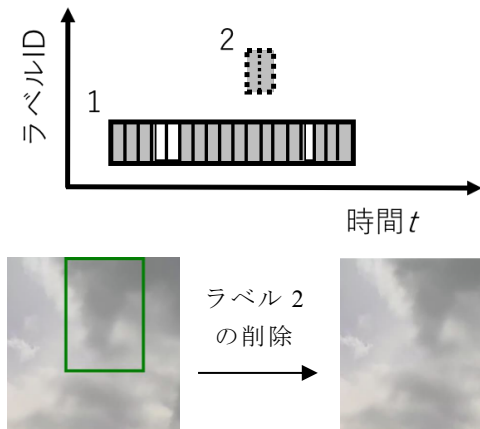


図7 ラベル付きオブジェクト候補リストの更新期間(上)と一時的に発生する誤検出解について更新回数がしきい値以下の事例(ID 2)を削除した事例(下)

4. 実験

検証実験は検出部と追跡部を分けて実施した。使用した動画は 2017 年 9 月 11 日に高知県安芸市に南西方向、南東方向に向けて設置された気象監視カメラ(それぞれ ch1、ch3)で撮影されたフルハイビジョン、1fps のカラー動画である。画像枚数はそれぞれ 1035 枚、1188 枚、含まれる事例(漏斗雲)数は 1035、1088 である(実体としては 2 つである)。データ数が限られるため学習・検証には、データセット 1 では ch1 を学習、ch3 を検証に使用したケース、その逆をデータセット 2 として利用した。

実行環境は CPU Intel Core i7 6700K (4.0GHz)、メモリ 32GB、GPU NVIDIA GeForce GTX 1080 (8GB)を搭載した Ubuntu 14.04 LTS である。

4.1. 検出性能の検証

性能の評価には精度、再現率を用いた。正解 $\{S_i | i = 1, \dots, im\}$ と推測解 $\{M_j | j = 1, \dots, jm\}$ に対して、下記のとおり推測解のうち正解と一致するもの、正解のうち推測解と一致するものをラベル付する。

$$P(S_i|M_j) = \begin{cases} 1 & \text{if } \max_i \frac{\text{Area}(M_j \cap S_i)}{\text{Area}(M_j \cup S_i)} > pth \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$P(M_j|S_i) = \begin{cases} 1 & \text{if } \max_j \frac{\text{Area}(M_j \cap S_i)}{\text{Area}(M_j \cup S_i)} > pth \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

そしてしきい値を超えたものを数え上げて、それぞれ検出領域の数、モデルの個数で割ることにより精度、再現率を求めた(実際には時間方向にも積算)。

$$\text{精度} = \frac{1}{jm} \sum_{j=1}^{jm} P(M_j|S) , \text{再現率} = \frac{1}{im} \sum_{i=1}^{im} P(S_i|M) \quad (4)$$

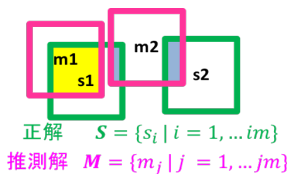


図 8 評価方法。正解と推測解の重複した領域(青、黄)の組から IoU がしきい値以上(黄)に該当するものをラベル付けして精度、再現率で評価。

図 9 に両手法での検出例を示す。YOLO に階層化画像を組み合わせる手法では誤検出が多いが Tiny YOLO v3 では少なく、一方、画面右側の旗のような障害物より隠蔽には、YOLO に階層化画像を組み合わせる手法の方が頑健であることがわかる。

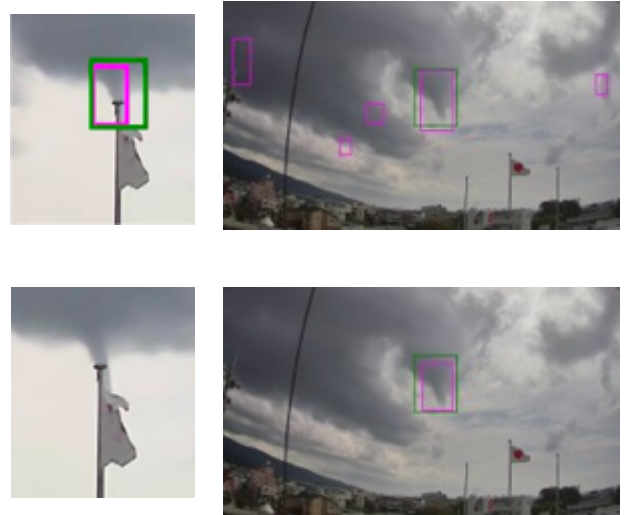


図 9 YOLO を用いた階層化アプローチでの検出結果(上)と Tiny YOLOv3 を用いた非階層化アプローチによる検出結果(下)。マゼンタの矩形は抽出結果、緑の矩形は人間の視覚で漏斗雲と判定された領域。

図 10 には各手法の精度と再現率を、検出時の IoU のしきい値の関数として示す。YOLO に階層化画像を組み合わせる手法は前述の通り隠蔽に強いことから再現率が高く精度が低い結果となった。一方 Tiny YOLO v3 では精度は高いものの大きく映った漏斗雲や障害物付近に存在する事例での検出失敗により再現率が低い結果となった。本システムでは災害感知という目的があるためさらに高い精度、再現率が求められることから YOLO に階層化画像を組み合わせる手法の方が好ましいと考えられる。また、IoU のしきい値については、 $pth = 0.2$ のとき結果が安定していることが確認した。これは正解データとして与えた領域とのマッチングに差があり、正解の設定の曖昧性の問題でもあると考えられる。

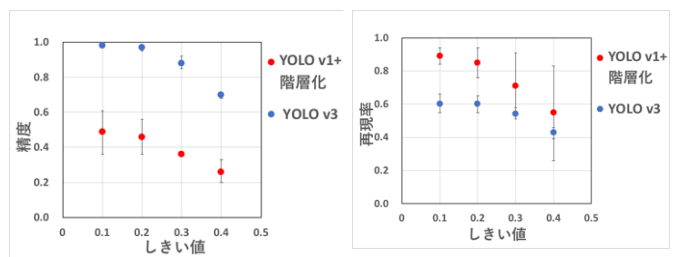


図 10 漏斗雲の検出評価結果。エラーバーはデータセット 1,2 を用いた際の各指標での最大値、最小値。

また表 1 に 1 フレームあたりの認識にかかる時間を示す。深層学習のネットワークを読み込む処理がボトルネックになっていることが確認できるが、ある程度画像を蓄積した上でまとめて処理するといった手法により計算時間の問題への対処は可能である。

表 1 認識に要した処理時間

手法	YOLO	Tiny YOLOv3
モデルの読み込み[s]	2.85	1.52
モデルの読み込みを除いた 1 枚あたりの処理時間[s]	2.03	0.133
1 枚あたりの処理時間[s]	4.88	1.65

4.2. 同一性判定の検証

同一性判定の検証には、4.1 節で述べたデータセット 1(安芸 ch3)について各手法で検出された検出解を利用した。この事例では目視で 2 つの漏斗雲が確認される。追跡を行う際のハイパーパラメータについては試行錯誤により、検出解と追跡候補とのマッチングを行うしきい値 IoU_{th} を 0.3, 追跡断絶の猶予時間のしきい値 T_n を 5, 追跡回数のしきい値 t_s を 100 とした。

図 11 には検出されたオブジェクトの最小包含矩形の面積を時間の関数として、同一性判定の結果つけられたラベルごとに異なる色で示す。ここで灰色はノイズと判断されたオブジェクトを示す。YOLO を用いた階層的アプローチを利用した場合には長期間の追跡が実現できていることが確認できるが、この中で真の解は、シアン、オレンジ、赤のグループが真の解であるが、オレンジ、赤のグループは本来 1 つのオブジェクトであるものが隠蔽によって統一できなかったものである。またそれ以外にもノイズを検出しているが、これは長期にわたり同じ箇所に誤検出解が存在することもあるためである。一方、Tiny YOLO v3 を用いたケー

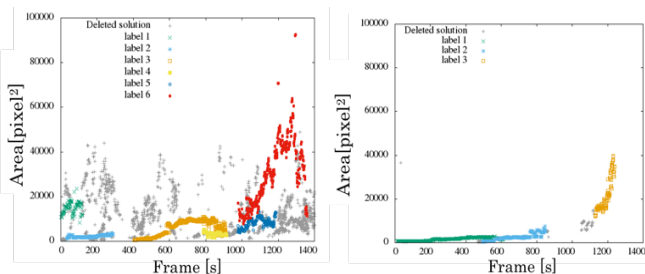


図 11 YOLO を用いた階層化アプローチでの検出結果を用いた追跡結果(左)と Tiny YOLOv3 を用いた非階層化アプローチの検出結果を用いた追跡結果(右)。同色の点は同一オブジェクトとして判定された事例、灰色の点は追跡の際に誤検出解として判定された事例。

スはもともとノイズが少ないため誤検出解は発生していないが、隠蔽による追跡の断絶期間(800sec 付近)がより長くなっている。この時の検出オブジェクトの様子を図 12 に示しておく。



図 12 Tiny YOLO v3 の検出解を用いたオブジェクト抽出と同一性の判定例。上は 750 フレーム、下の図は 750、857、1000、1120 フレーム。

このような同一性判定を取り入れることでオブジェクトの移動解析が可能になった。特に面積の拡大、縮小によって、接近、遠ざかりを判定することは災害現象のアラートینگのために重要である。

5. 可視化システムの実装結果

最後に 3, 4 章の結果をサーバに保存して Web からリアルタイムに参照することが可能なアラートینگ、アーカイブシステムを設計しプロトシステムを構築した。サーバ内にはリアルタイムで検出した結果を保存しており、ユーザがアクセスした際には最新の処理結果を確認することができるものとする。

アラートینگシステムを Web アプリケーションとして構築するにあたっては、サーバ内に保存されている最新のデータをユーザが意図的に更新しなくても参照できることが必要となる。そこで本システムにおいてはポーリングにより最新のサーバ内の情報を取得させることとした。ポーリングはユーザがアクセスしている Web ページから一定時間ごとにサーバにアクセスさせることで最新のデータを取得する方法である。図 13 にポーリングを用いたリアルタイム表示システムの概念図を示す。Web ページの処理は次のように行われる。

- ① ユーザが Web ページにアクセス
- ② サーバ内の HTML ファイルをユーザにレスポンス
- ③ ①で取得したファイルに埋め込まれた JavaScript によりサーバの PHP ファイルにアクセス
- ④ 最新の検出結果を PHP ファイルに埋め込みユーザにレスポンス
- ⑤ ④で取得した PHP ファイルを HTML ファイル内に埋め込んで表示
- ⑥ ③-⑤を一定時間おきに繰り返す

なお、サーバ内に保存されているファイルを読み込む処理は PHP で実装し、処理結果の描画やポーリング処理については JavaScript のライブラリである jQuery も利用して実装した。実装に用いたのはそれぞれ、Apache 2.0、PHP 7.3、jQuery 3.4.1 である。

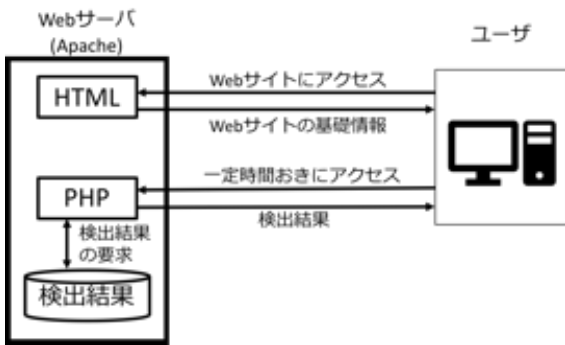


図 13 ポーリングを用いた本システムの実装

図 14 に構築されたプロトタイプシステムを示す。左側の表示はカメラサイトの設置点と漏斗雲検出状況を示している。ここでは検出に用いられているサイトの地点を丸で示しており、平常時では黒丸、漏斗雲が検出されたサイトでは赤丸で描画する。右側では最新の画像における漏斗雲検出結果を表示する。ドロップダウンリストからサイトを指定して表示を押すことで、指定したサイトの 4 方向の画像のリアルタイム検出結果を表示することができる。各サイト画面では漏斗雲が観測された際には漏斗雲の箇所を赤枠で表示する。これらの結果については前述のポーリングにより一定時間おきに更新されているためリアルタイムで漏斗雲発生状況を確認することが可能である。

6. おわりに

本研究では高知大学のカメラで取得された動画像を用いて深層学習による漏斗雲の自動検出と追跡を行った。そしてその結果について Web アプリケーションを介してリアルタイムで確認できるようなシステムを

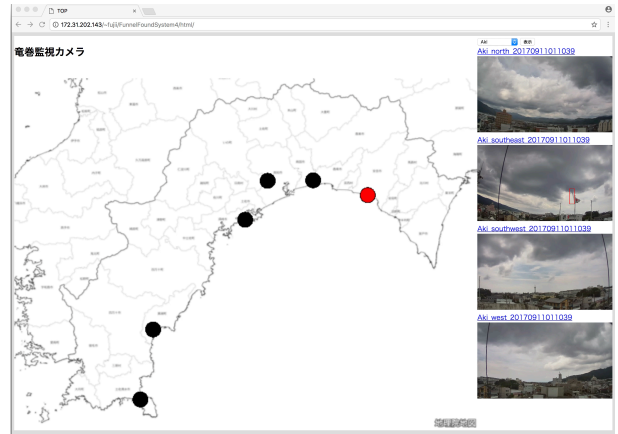


図 14 可視化システム。左は広域の可視化。サイトの設置点は丸で示しており、黒が平常、赤は漏斗雲が検出されたサイトとしている。右は指定したサイトでの漏斗雲検出結果を描画。地図は[5]より。

構築した。漏斗雲の検出・追跡については深層学習のオブジェクト検出手法である YOLO を用いて検出を行なったのちに、検出された解をもとに隣接した時間の解で近傍に発生したものが同一解と判定することで追跡を行なった。この手法では小さな事例も含めて連続的に検出・追跡できることを確認したが、障害物に隠れるような事例では長期間検出が失敗してしまい追跡も行われなかったことが確認できた。Web アプリケーションによる表示についてはリアルタイムでの結果表示ができることが確認できた。今後の方針としては検出・追跡手法の洗練化を行い、検出・追跡ができなかった事例の検出。またユーザに提供する結果についても生命線や方角の提示、漏斗雲のテキストチャのクラスタリングなどによる自動解析。さらには大量データ、多人数アクセスした際に各処理が想定トラフィック、処理時間で処理できるかも含めて検討していきたいと考えている。

謝辞

本研究は科研費 17K00158 および 18H01682 の補助により進められました。ここに感謝いたします。

参考文献

- [1] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., You only look once: Unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.779-788, 2016.
- [2] Redmon, J., and Farhadi, A., YOLOv3: An Incremental Improvement, arXiv preprint, arXiv:1804.02767, 2018.

- [3] 藤井祐貴, 檜垣啓汰, 中山慎也, 本田理恵, 佐々浩司, ネットワークカメラを用いた災害情報マイニングの試みー深層学習による竜巻とその前兆状態の検出ー, DEIM Forum 2018, K2-2, pp.1-8, 2018.
- [4] 佐々浩司, 西井章, 2017年9月11日に高知県沿岸部で発生した複数の竜巻, 日本気象学会関西支部第1回例会公演要旨集, 142, pp.9-12, 2017.
- [5] 国土地理院, <https://cyberjapandata.gsi.go.jp>, (2020年1月6日閲覧)