

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 — 深層学習による竜巻とその前兆状態の検出 —

藤井 祐貴[†] 檜垣 啓汰[†] 中山 慎也[†] 本田 理恵[‡] 佐々 浩司[‡]

[†] 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

[‡] 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

E-mail: [†] [‡] {b143k281, b143k199, b143k174, honda}@is.kochi-u.ac.jp, [‡] sassa@kochi-u.ac.jp

あらまし 近年、情報通信技術の発達により、遠隔地の映像を収集、保存、分析することが容易になってきている。竜巻のような災害に関連した現象も、従来はたまたまその場で遭遇した人によって映像や写真が残されることが主流であったが、現在ではネットワーク化されたカメラによって広域かつ定常的に監視を行うことが可能になりつつある。このようなネットワーク化された気象監視カメラから竜巻とその前兆状態を自動的に検出するための手法の開発を行った。竜巻検出では深層学習の1つである畳み込みニューラルネットワークに基づいた2手法として、Yoloによるオブジェクト領域推定と認識、ブロック単位のクラス推定を検討した。また、前兆状態の検出には雲の動きから見られる気流の分布に注目することで判定できると仮定し、オプティカルフローを用いた気流検出に基づく手法を検討した。

キーワード 災害時監視支援システム, 機械学習, 深層学習, オプティカルフロー

1. はじめに

近年、情報通信技術の発達により、遠隔地の映像の収集、保存、分析が容易になってきている。気象や災害に関連した分野でも、竜巻や川の水位、土石流の監視、火山の噴煙監視など様々な映像情報が活用されている。このような災害に関連した現象は、従来、偶然その現象に遭遇した人によって残されることが主流であったが、現在ではネットワーク化されたカメラによって広域かつ定常的な監視を行うことが可能になりつつある。これらのリアルタイムセンシングデータと機械学習の手法を組み合わせる事によって、災害をもたらす現象の予兆、発生、状態変化を検知し、災害に関する情報をマイニング（発掘）して、避難情報の提供などに活用できる可能性がある。

高知大学では、全国的にも高い頻度で発生する”竜巻”の監視のために、土佐湾周辺に12基のネットワーク化されたモニタカメラを設置している^[1]。取得した画像は現在目視で閲覧しているが、モニタカメラから竜巻の初期状態を示す”漏斗雲”(漏斗状の雲)を自動的に抽出することができれば、直前予測、アラートに活用することができる。また、この情報を網羅的にデータベース化することによって、研究者による典型的な移動パターンやその発生メカニズムの解明に役立てる事ができる。

本研究では、このようなリアルタイムの災害情報マイニングシステムのコアプロセスとして、実際に高知大学の気象監視カメラで取得された映像を用いて、深層学習による異なるスケールの漏斗雲の自動検出と、オプティカルフローを用いた気流の状態変化検出による前兆状態としての雲行きの悪さの判定の試みについ

て報告する。

2では想定している竜巻とその前兆現象を検出するシステムの概要についてまとめ、3では深層学習を用いた竜巻の自動抽出手法について述べる。4では問題を前兆現象の検出にしばってオプティカルフローの分布から機械学習で雲行きの悪さを判定する手法について述べる。最後に5で今後の課題等についてまとめる。

2. システム構成

まず本研究の対象は土佐湾を囲む高知大学朝倉キャンパス、物部キャンパス、安芸の3カ所にそれぞれ4方向に向けて配置された広角監視カメラで1秒間隔に取得された画像とする。図1に2017年9月11日に発生した竜巻をこのカメラでとらえた例を示す。ここでは、雲底から突出した漏斗雲が観測点に近づき、通り過ぎる様子が克明にとらえられている。



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。左右のフレームの時間間隔は215 sec.

本研究で構築するシステムの概念図を図2に示す。リアルタイムでネットワークを通じて取得される気象監視カメラの画像から、竜巻の前兆現象である漏斗雲の自動抽出を行い、アラートに活用し、データ

ベース化する。またデータベースに保存された情報から機械学習を行い、新たな知識発見を促すシステムを構築することを想定する。

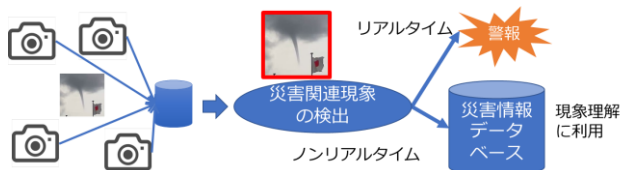


図2 竜巻・漏斗雲判定システム

3. 深層学習を用いた漏斗雲の抽出

まず画像から漏斗雲，竜巻を機械学習の手法によって自動的に抽出する手法を検討する。漏斗雲，竜巻は共通した円錐状の形状をもつが，その境界面はあいまいで，形状も不定であり，また背景との明暗の関係が逆転していることもあるため，幾何学的なモデルを設定することは難しい。このため観測された事例から，モデル構築を行う深層学習の手法を検討した。

深層学習は深いニューラルネットワークを利用した手法の総称であり，なかでも畳み込みニューラルネットワーク(Convolutional Neural Network: CNN)は画像認識の分野で実績がある。物体の抽出手法として本研究ではオブジェクト抽出と分類を同時に行う手法とブロックのクラス分類を行う手法の2手法を検討した。

3.1. オブジェクト抽出と分類による手法

3.1.1. 手法

深層学習を用いた物体の抽出，認識を同時に行う手法として，R-CNN (Region with CNN)^[2]，Fast R-CNN^[3]が知られているが，動画像処理も想定して，高速に物体の抽出と分類を同時に学習する事が可能な Yolo (You only look once)^{[4][5][6]}を検討する。

Yolo は C 言語で実装されたニューラルネットワーク用のフレームワーク Darknet^[7]の一部である。24 層の畳み込み層と 2 層の全結合層から構成されているが，入力層では固定サイズの入力画像を $S \times S$ のセルに分けて，セル単位で，オブジェクト領域(B 個)を示す 5 つのパラメータ(中心座標，幅，高さ，confidence)，オブジェクトのクラス確率を設定し，ネットワークが予測した候補領域が学習データと一致するよう重み係数の学習を行う。この手法を用い，インターネット上の web サイトから収集した静止画や観測者からの提供画像から学習した結果の実画像への適用性について調査した。またここでは”竜巻または漏斗雲”という 1 クラスの抽出問題として実験を試みた。

3.1.2. 予備実験

実験に用いたデータを表 1 に示す。インターネット

上でから「funnel cloud」，「tornado」，「water spout」のキーワード検索によって収集した画像と国内の観測者から提供された画像の合計 101 枚を使用した。ここから視覚的に 106 の竜巻，漏斗雲を抽出してデータセットとした。画像のサイズには縦横比も含めて，様々なものが混在している。

表 1 実験に用いたデータ

	画像枚数	オブジェクト数
インターネットから収集	86	90
国内で撮影	15	16
合計	101	106

学習の際には，試行錯誤から検出の閾値は 0.2，学習のエポック数は 2000 とした。入力時に画像は固定サイズ(448 pixel×448 pixel)にスケーリングされた後 $7 \times 7 = 49$ セルに分割される。セル毎の抽出候補となるオブジェクト領域数 B は 3 とした。

検証には 5 回交差検定を行った。データセット 1 つあたりのテストデータのオブジェクト数は 21，学習データのオブジェクト数は 85 であった。評価の指標は，再現率(ターゲット事例のうち，実際に抽出されたものの割合)，精度(ターゲット事例として抽出されたもののうち正しい物の割合)，F 尺度($2 \times \text{再現率} \times \text{精度} / (\text{再現率} + \text{精度})$)を用いた。

実行は CPU Intel Core i7 6700K (4.0GHz)，メモリ 32GB，GPU NVIDIA GeForce GTX 1080 を搭載した ubuntu 14.04 LTS のマシンで，CUDA による GPU モードで行った。5 回交差検定の各ケースの学習には約 3 時間を要し，また学習したモデルを用いたテスト事例に対する検出時間は 1 枚あたり 2.6 秒程度であった。

表 2 に 5 回交差検定の平均的な結果を示す。この実験から精度は 0.964 と高いものの，再現率は 0.820 と低めの値であることがわかる。5 回の実行例のなかでは再現率が 0.714 まで低下するケースもあり，この場合，最大で約 30% の事例を見落とす可能性があることになる。

表 2 5 回交差検定を行った結果

	再現率	精度	F 尺度
平均	0.820	0.964	0.884

図 3 には成功事例，失敗事例を示す。検出失敗の主な原因としては，信号や噴水，水滴など類似の形態を

もつ障害物が大きく写っていること、竜巻や漏斗雲の周辺に雲が多く存在していることの2つがある。学習時と予測時にリサイズを行っているため、画像の縦横比によっては竜巻や漏斗雲とポール等の障害物が判別不可能になっていることも考えられる。

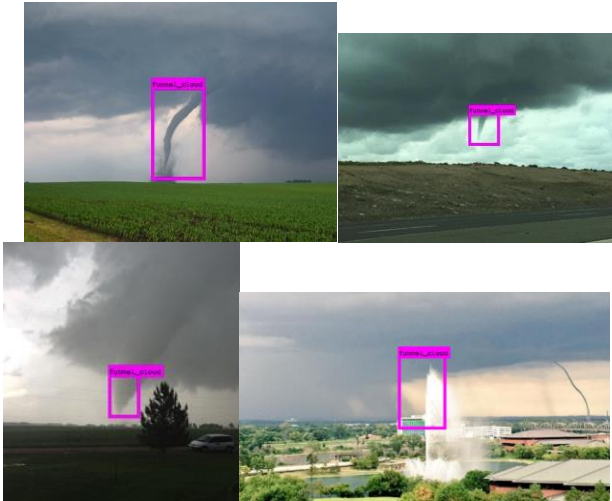


図 3. 検証事例での成功例と失敗例^{[8][9][10][11]}. マゼンタの枠が抽出された漏斗雲候補。

また、この学習結果を用いて、実際の一連の動画像のフレーム画像に対しての竜巻抽出実験もおこなった。その一例を図 4 に示す。この結果から、学習事例として与えられたような視認しやすい竜巻の抽出には成功し、また一旦検出に成功すると、時間的发展はモデルとして考慮していないにもかかわらず、時間とともにあたかも追跡するように竜巻を抽出し続けることができることを観察した。しかし、図 4(上) に含まれるやや障害物付近の漏斗雲については抽出に失敗し、遠方の微小な竜巻については一律に抽出に失敗した。

これは固定サイズを前提とした入力層にスケールした画像を与えているためであり、特に入力時に、横に伸長した画像を強制的に正方形にスケールしたために、縦方向に伸長した細いパターンの特徴が失われて、検出されなくなったものと考えられる。災害関連現象の検出としては、小さい特徴から大きな特徴までシームレスに検出することが重要であるため、学習事例の適切な選択、スケール、入力画像の分割化、階層化などによる対策が必要である。

3.1.3. 微小漏斗雲検出のための改良

3.1.2 の結果をふまえて小さい漏斗雲の検出を目標とした追加実験を行った。ここでは画像全体を入力とせず、Yolo の入力ユニット数と同程度の画像にブロック化(1920 pixel×1080 pixel の画像を 8 ブロックに分割)して、極端なスケールをすることなく学習、認



図 4. 学習結果を用いた実画像(2017 年 9 月 11 日、安芸 ch3(南東方向))での認識例。マゼンタの枠が抽出領域。上図は検出失敗、下図は検出成功例。

識が行えるように調整を行った。また小さい漏斗雲を実画像からサンプリングしてデータ拡張を行って学習データにすることによって、微小な漏斗雲の学習事例に対する強化を行った。

実験に使用したデータを表 3 に示す。この実験ではインターネットから取得したデータではなく、実際の監視カメラで撮影された画像を実験データに用いた。なお、同じサイトの画像を学習と評価の両方に使用しないように、分割して使用するものとした。

表 3 実験に用いた元データ

	画像枚数	オブジェクト数
安芸 ch1(南西方向)	25	29
安芸 ch3(南東方向)	30	32
物部 ch2(東方向)	9	9
合計	64	70

表 3 のデータ数では学習事例数として不足のため、データ拡張を行った。まず、監視カメラ画像から漏斗雲を中心として、どこを中心としても漏斗雲の包含正方形となるような中心位置を取る範囲を決めた。そして、ランダムに中心位置をずらしながら 3 種類のサイズの正方形を 5 種類サンプリングする。このデータ拡張の結果、表 4 に示す通りサンプリングされた画像枚数は 960 枚となり、安芸 ch1 に物部 ch2 を加えたものと安芸 ch3 の 2 種類のデータを学習データと評価データにそれぞれわけて使用した。学習は 1 回のみとしたが、評価には異なるサイトの画像を使用することにした。

検出の閾値は試行錯誤より 0.1 にし、学習エポック数を 2000 とした。

表 4 拡張後のデータ

	画像枚数	オブジェクト数
安芸 ch1 (南西方向)	375	404
安芸 ch3 (南東方向)	450	460
物部 ch2 (東方向)	135	135
合計	960	999

表 5 に実験結果の性能を示す。また、図 5(a)に検出成功事例、図 5(b)に検出失敗事例を示す。予備実験時より実動画での精度、再現率が上昇し、実画像においても図 5(a)に示されるような遠方の小さい漏斗雲を比較的安定的に抽出する事ができるようになった。一方で、大きな漏斗雲、竜巻は検出しにくくなったため、画像を階層化して、様々なスケールのオブジェクトを並列に処理出来るようにする事が必要である。このように入力画像のスケールに対して配慮を行いながらブロック化し、また学習データについてもデータ拡張によって性能が向上したが、計算時間は1桁程度増大する事になった。これについては 3.2 で述べるような事前の選別が有効であると考えられる。また図 5(b)のように境界部でのオブジェクト抽出が不連続になるという問題も生じるが、階層化レベルの高い画像やオーバーラップをもたせたブロックの利用で対応することが考えられる。

表 5 追加実験の結果

学習	検証	精度	再現率	F 尺度
安芸 ch1	安芸 ch3 (データ拡張)	0.844	0.896	0.869
	安芸 ch3 (実動画)	0.556	0.717	0.627
安芸 ch3	安芸 ch1 (データ拡張)	0.913	0.636	0.750
	安芸 ch1 (実動画)	0.641	0.693	0.666

3.2. ブロックのクラス分類

3.2.1. 手法

前節の Yolo をもちいた CNN によるオブジェクト抽出手法では、発生してすぐの小さな漏斗雲の検出には画像の階層化に伴う計算時間の増大という問題があった。この課題に対する異なるアプローチとして、画像を小領域のブロックに分割してブロックごとに”漏斗



図 5 評価データへの適用例. (a), (b)共に 2017 年 9 月 11 日安芸 ch3(南東方向)の監視カメラ画像

雲が含まれるか否かを CNN を用いて”クラス分類を行うことで、微小な漏斗雲の検出を試みる。その際、この処理を全画像に適用すると計算時間が増大してしまうため、小さい漏斗雲は雲底に発見されることを利用して、特徴抽出フィルタを適用して漏斗雲が発生しそうな雲底部分を大まかに発見した後に CNN を適用することを考える。

具体的な手順は以下の通りとした。

- (1) 漏斗雲が撮影された画像を 64 pixel×64 pixel のブロックに分割
- (2) 雲底抽出のためのフィルタを適用し、フィルタの出力の総和が閾値以上になったものを雲底ブロックとして判定
- (3) 雲底と判定したブロックごとの漏斗雲ブロック、漏斗雲でないブロック(2 クラス)のクラス分類を CNN で学習させることで、漏斗雲と漏斗雲以外の雲底を判定

雲底は暗い雲から明るい空の境界線が水平方向に存在する領域で、ある程度の連続性を持つという特徴から、図 6 のような 5×5 サイズの横方向の縁を抽出するフィルタを使用した。フィルタリングの結果が 50 以上となるものは雲底画素であると判定し、さらに 64 pixel×64 pixel ブロック内に 16 pixel 以上雲底画素と判定されたピクセルがある場合、雲底があるブロックと判定するようにした。

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
0	0	0	0	0
1	1	1	1	1
1	1	1	1	1

図 6 雲底判定様の 5×5 のフィルタ

3.2.2. 畳み込みニューラルネットワーク

ブロックのクラス分類には VGG-16^[12]を用いた。VGG-16は畳み込み層 13 層，プーリング層 5 層，全結合層 3 層からなるモデルで，最後にソフトマックス層によってクラス分類がなされる。ニューラルネットワークのフレームワークとしては Theano. API として Keras を用いている。出力はソフトマックス層により 0 から 1 の実数値となるが，閾値処理を行うことで漏斗雲かそうでないかの判定を行った。

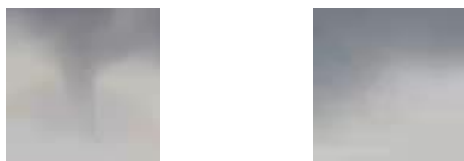
3.2.3. 実験データ

図 7 に学習と検証に用いた画像(カラー，1920 pixel × 1080 pixel)の一例を示す。ここでは安芸 ch1(南西方向)，安芸 ch3(南東方向)の気象監視カメラから取得された画像を用いている。ここから漏斗雲を含むブロックと含まないブロックをそれぞれ 824 個ずつ抽出して 224 pixel × 224 pixel にスケーリングした。



図 7 使用した画像の一例

漏斗雲については肉眼で観測されたものをサンプリングして用いた。その際，データ拡張のため，サイズについては漏斗雲の最小包正正方形，その 1.2, 1.5 倍の通りの領域をサンプリング，スケーリングして使用した。また位置のバリエーションをもたせるため，中心位置をランダムにずらしたものをそれぞれのスケールについて 5 通りサンプリングした。一方，漏斗雲でないブロックについては気象監視カメラの画像を，64 pixel × 64 pixel の大きさのブロックに分割し，3.2.1 で述べた手法で雲底と判定されたブロックからサンプリングした。この際，漏斗雲，建造物を含むブロックは対象から除外した。このようにして抽出されたブロック画像の例を図 8 に示す。



(a) 漏斗雲のブロック (b) 漏斗雲でないブロック

図 8 作成したブロック画像の一例。(a), (b)はそれぞれ図 7 の赤枠，緑枠領域に相当

3.2.4. 実験

実験に用いたデータは表 6 のように学習データと評価データ，テストデータの 3 種のデータセットを作成した。なお，学習データ，評価データは安芸 ch1(南西方向)，テストデータは安芸 ch3(南東方向)の気象監視カメラから作成している。

表 6 データセットの内訳

	漏斗雲 ブロック	非漏斗雲 ブロック
学習データ	218	218
評価データ	217	217
テストデータ	389	389
合計	824	824

過学習を避けるため，ホールドアウト検証を行い，エポック数は学習を行った中で評価データに対して推測を行ったとき，一番正解率が高いモデルを用いた。評価の指標としては True Positive Rate と False Positive Rate を用いた。True Positive Rate は漏斗雲のブロックを正しく予測した割合，False Positive Rate は漏斗雲でないブロックを正しく予測した割合である。

図 9 に実験によって求められた True Positive Rate, False Positive Rate を漏斗雲と判定するための閾値の関数として示す。閾値を上げることによって，False Positive Rate, True Positive Rate のいずれも減少していくが，False Positive Rate の方が急激に立ち下がるため，高い認識性能が達成できる認識器が形成出来ている事がわかる。ただし，同じ画像から漏斗雲のブロックに関しては拡張処理を行ったため，過剰に学習を行い，性能が高まったように見える可能性もある。

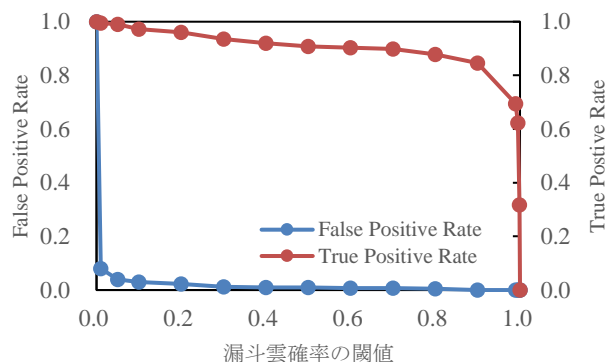


図 9 VGG-16 による漏斗雲，非漏斗雲ブロックのクラス分類における，閾値の変化させた場合の True Positive Rate と False Positive Rate

また，この学習結果を用いて，実際の一連の動画画像のフレーム画像に対しての竜巻抽出実験もおこなった。

図 10 に検出結果の可視化例を示す。ここで矩形の領域は雲底を含むと判定されたブロックで、その枠の色は VGG-16 で求められた漏斗雲を含む程度を示す。図 10 では右端に漏斗雲度が 0.875 以上（赤）の領域が存在し、ここには確かに遠方の小さい漏斗雲が含まれている。



図 10 VGG-16 による漏斗雲ブロックの抽出結果例

図 11 に推測結果の成功事例と失敗事例のクローズアップ画像を示す。図 11(左)のように漏斗雲の全体がブロックに含まれている場合には漏斗雲である確信度は大きくなるが、図 11(中央)のように漏斗雲がブロックの境界上に存在する場合は小さくなる傾向があることがわかる。また図 11(右)のように漏斗雲でないブロックも局所的な特徴から高い確信度を示してしまっていることが確認できる。こうした場合の精度もふくめて精度向上させるには、周囲のブロックの状態も視野に入れながら判定するような方式に改良する必要がある。

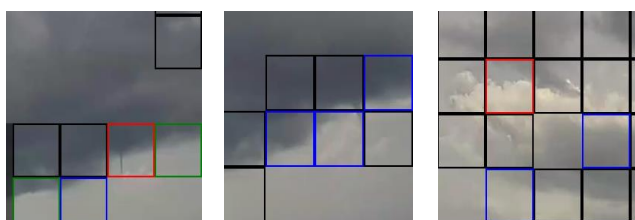


図 11 学習結果を用いた実画像(2017年9月11日、安芸 ch3(南東方向)での認識例。ピックアップ処理をした。左: 検出成功例, 中央: 抽出失敗例, 右: 誤検出。色の指標は図 10 のカラースケールバーと同じ。

表 7 に閾値を変化させたときの分類性能を示す。この実験からは閾値が 0.875 のときに F 尺度が一番高く、かつ再現率も立ち下がっていないためバランスが良いと考えられる。再現率は最低でも 0.8 程度と高いのに対し、精度は 0.388 程度と低い傾向である。

表 7 ブロック単位の漏斗雲分類の性能

閾値	精度	再現率	F 尺度
0	0.010	1.000	0.019
0.25	0.054	0.955	0.102
0.5	0.138	0.914	0.239
0.75	0.331	0.809	0.470
0.875	0.388	0.801	0.523

3.3. 両手法の比較

ここまで検討した全ての手法の実動画に対する漏斗雲の検出性能と計算時間の比較を表 8 に示す。精度は Yolo によるオブジェクトの抽出と分類を行う手法が 0.599 とやや高く、再現率は VGG-16 によるブロックのクラス分類の手法が 0.801 とやや高い結果となった。竜巻発生のアラートに活用する場合においては災害情報として見逃さないということが重視されることを考えると、再現率が高いブロック分類が有利であると考えられる。一方、リアルタイム性からは微小漏斗雲を対象とすると 1 枚あたり 30sec 程度の処理時間が必要で、その間には雲の状態が様変わりするため、5-10 倍程度の高速化は必要であると考えられる。これについてはネットワークの簡素化、学習済みのモデルの読み込み時間の短縮、分散化や並列化が必要であると考えられる。

表 8 漏斗雲抽出性能の手法間比較

手法	精度	再現率	F 尺度	学習(分)	推測 (s/frame)
Yolo (閾値 0.1)	0.245	0.216	0.272	543	2.5
Yolo 改良* (閾値 0.1)	0.599	0.705	0.647	543	25
VGG-16 (閾値 0.875)	0.388	0.801	0.522	19**	35**

*表 5 安芸 ch1, ch3(実動画)の平均値

**若干異なる環境 (CPU Xeon E5-1620v4 (3.5GHz), メモリ 16GB, GPU NVIDIA GeForce GTX 1080 Ti, NVIDIA Quadro P400, OS ubuntu 16.04 LTS) で実行

4. オプティカルフローによる前兆状態判定

4.1. 手法

これまでは漏斗雲が発生してからの抽出に焦点をおいたが実際には発生以前の予兆的現象を検知するのも災害情報マイニングの観点からは重要である。ここでは漏斗雲が発生する前兆現象の検出可能性について検討する。

竜巻のこの前兆現象としては気流の状態（雲の移動速度）の変化が使用できる可能性がある。雲行きが悪さは上層と下層で異なる方向に雲が移動している、遠方で渦状の気流が観測される等から判定できる可能性

がある．雲行きが悪さをオプティカルフローによる移動方向ベクトルの見かけの方位角分布から検出することを試みた．

手法の概略は以下の通りである

- (1) 10 秒程度の短い時間間隔の 2 枚の画像からオプティカルフローとして気流の速度ベクトルを求める．
- (2) 速度ベクトルの方位角をもとめ、その画像内の頻度分布を求める
- (3) EM アルゴリズム^[13]を用いて、頻度分布を混合ガウス分布でモデリングすることで、類似の移動方向をもつ気流の成分を求める（上層下層で別方向に移動している様子をカメラがとらえて入れば 2 極分布になると予想される）．この際、ベイズ情報量規準(Bayesian information criterion : BIC)^[14]により雲の成分数の自動判定を行う
- (4) 類似の移動方向の成分数の時系列変化から雲行きの変化を推測する

なお、ここでは(4)の類似速度をもつ成分数の時系列変化の導出までを行なった．

4.2. 実験

実験に用いたデータは 3 で使用しているものと同じ高知大学の気象監視カメラの画像である．竜巻通過 30 秒前(前兆時)と竜巻通過 80 秒後(平常時)の画像に対してオプティカルフローの付与を行った．オプティカルフローはステレオマッチングと同様のブロック単位でペア画像を走査しながら差分の 2 乗和を最小にするブロックをもとめ、移動量を速度ベクトルに比例する値とする方式でもとめた^[15]．

オプティカルフローを付与するにあたってのパラメータとして、移動方向を求めるための注目ブロックを決めるウィンドウサイズ、画像内を走査する場所を決める走査範囲、次のオプティカルフローを付与するためのウィンドウと走査範囲のズレ幅のステップ数を与えた．また特徴の無いブロックは対応付けが困難であることから、ブロックマッチングを実施するか否かについて、画像内の輝度の標準偏差に閾値を設け、閾値を超えたものだけでマッチングを行った．表 9 にこれらのオプティカルフローの実施条件を示す．

図 12 上に抽出された竜巻通過 30 秒前のオプティカルフローの結果を示す．雲の移動方向とオプティカルフローの向きが対応していることが確認できる．また領域毎に速度が大きく異なるのも観察できる．しかし、偽対応による大きな速度ベクトルのノイズが所々発生しており、特に平常時において多くなっている．一方、

表 9 オプティカルフロー実施条件

条件	値
時間間隔	10 秒 (10 フレーム)
ウィンドウサイズ	21 pixel × 21 pixel
STEP 数	20 pixel
走査範囲	-20 pixel - +20 pixel
標準偏差閾値	1

図 12 下は竜巻通過 80 秒後の状態である．ここでも雲の移動方向とオプティカルフローの向きが対応していることが確認できるが、通過前の状態と比べるとまとまりのない速度分布になっており、前後で気流の状態が様変わりしており、このような分布の変化が前兆状態の検知に使用出来る可能性がある．



図 12 オプティカルフロー事例, 2017 年 9 月 11 日, 高知県安芸 ch1(南西方向). 上: 竜巻通過 30 秒前, 下: 竜巻通過 80 秒後

図 13 にはこの速度ベクトルの頻度分布を、見かけの方向毎にレーダグラフとしてまとめた結果を示す．竜巻通過前の特徴として、速度 0 付近のピークを除くと雲の移動速度と移動方向から 2 方向の分布があるこ

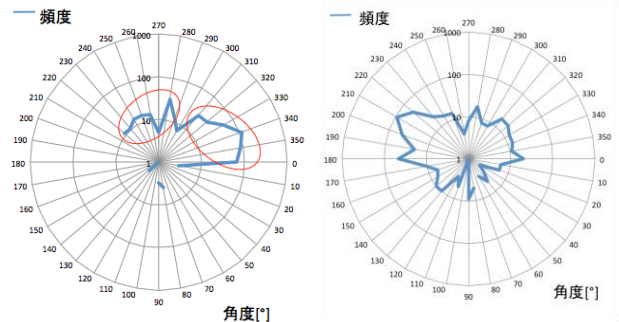


図 13 速度方向の頻度分布．左は竜巻通過 30 秒前(前兆時), 右は竜巻通過 80 秒後(平常時)

とを確認できる。これは竜巻の発生直前に上層と下層で異なる方向の気流が存在している事を意味している。

こうしたデータについて、EM アルゴリズムによる混合ガウス分布モデリングとベイズの情報量基準による成分数の自動判定を行った。図 14 には、EM アルゴリズムによって評価した雲の成分数が時間変化によって変化したものを示す。竜巻発生直前状態の特徴である漏斗雲が観測された時間から雲の成分数が大幅に増加していることが確認することができる。このことから、EM アルゴリズムを用いて竜巻前兆状態の検出することが可能であると考えられる。竜巻通過後において雲の成分数が竜巻通過前と変化がないが、これはオプティカルフローが竜巻通過後において偽対応により付与が失敗されることが多いために正しい雲の成分数が評価できていないことが原因であると考えられる。

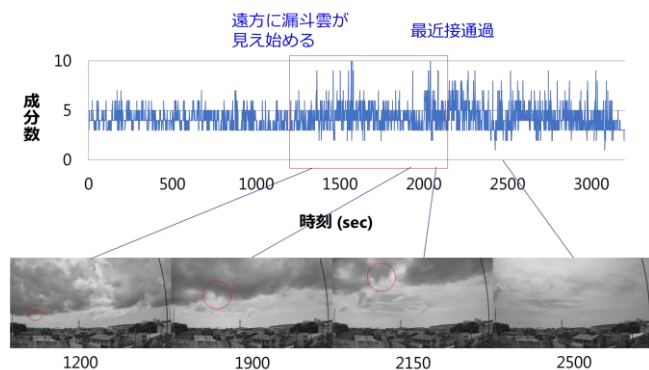


図 14 判定された雲の成分数の時間変化。上: 判定された雲の成分数，下: 各時間における気象監視カメラの画像。漏斗雲が発生している位置を赤枠で示している。

5. おわりに

本研究では実際に高知大学のカメラで取得された映像を用いて、深層学習による異なるスケールの漏斗雲の自動検出と、オプティカルフローを用いた気流の状態変化検知による前兆状態としての雲行きの悪さの判定を試みた。深層学習については、顕著な漏斗雲が実画像からも検出される事を確認したが、従来の手法では難しかった微小漏斗雲の検出についても、リアルタイム性を確保しながら実施する上での問題点と方針を確認する事ができた。オプティカルフローを用いた気流の状態変化については、偽対応発生によるノイズの問題があるが、気流の大まかな流れを示すこと、状態変化の検知に利用できる可能性があることが確認できた。今後の方針としては、各手法の洗練化を行い、これらを統合したセンサデータを用いた災害情報マイニングシステムの一例としての竜巻を含む気象監視カメラシステムの構築を検討していく予定である。

謝辞

本研究は科研費 15H02994 および 17K00158 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] Sassa, K., et al., "Characteristics of mesocyclones observed on Tosa Bay in Japan", Preprint of the 6th European Conference on Severe Storms, Palma de Mallorca, Spain, (2011).
- [2] Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation.", Proceedings of the IEEE conference on computer vision and pattern recognition, (2014), pp. 580-587.
- [3] Girshick, R. "Fast r-cnn.", Proceedings of the IEEE international conference on computer vision, (2015), pp. 1440-1448.
- [4] Redmon, J., et al., "You only look once: Unified, real-time object detection.", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016) pp. 779-788.
- [5] Redmon, J. and Farhadi, A., "YOLO9000: Better, Faster, Stronger", arXiv preprint arXiv:1612.08242 (2016)
- [6] 藤田一弥, 高原歩, 実装ディープラーニング, オーム社(2017) pp.266.
- [7] Redmon, J., "Darknet: Open Source Neural Networks in C", <http://pjreddie.com/darknet/>, (2013—2016)
- [8] NOAA National Severe Storms Laboratory, "Severe weather 101—tornado-", <http://www.nssl.noaa.gov/education/svrwx101/tornadoes/>.
- [9] Phi Plait, "Colorado Tornadoes", http://www.slate.com/blogs/bad_astronomy/2015/06/05/colorado_weather_tornadoes_in_boulder_county.html, (2015).
- [10] Kolotv.com, "Funnel clouds reported near Fernley", <http://www.kolotv.com/content/news/Funnel-clouds-reported-near-Fernley-see-photos-and-video-375197101.html> (2016)
- [11] TORNADO NEWS, <https://weather.com/storms/tornado/news/omaha-council-bluffs-skinny-tornado-photos>
- [12] Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv preprint arXiv:1409.1556 (2014)
- [13] Dempster, A. P., Laird, N. M., and Rubin, D. B., Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B(methodological), pp.1-38, 1977.
- [14] MacQueen, J., Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14), pp.281-297, 1967, June.
- [15] 長尾智晴, C言語による画像処理プログラミング入門, 朝倉書店(2014) pp.155.