

熊本震災ミュージアムDBにおける 画像認識Darknetによる震災画像への 自動タグ付けに関する研究

鬼気 拓海 廣重 法道 鶴田 直之

概要: 2016年(平成28年)に起こった熊本震災に関するデータを管理するために,本研究室で「熊本震災ミュージアムDB」というデータベースシステムを構築している.このDBには今後,多様・多量の情報を登録する予定である.それに伴い,データ利用に対するデータの体系化,及び効率的なデータ検索の方法の実現が重要である.本論文では,「検索タグ」機能の精度を向上させる目的として,Deep Learningによる画像認識処理を活用したタグの自動生成を行うための実験と,異なる特徴を持つ学習データから生成されるモデルに生じる検出精度の差を,再現率・適合率・正解率の観点から比較した.

キーワード: 地理データベース, 空間データ管理, マルチメディアデータベース

Study on Automatic Tagging of Earthquake Images by Image Recognition Darknet in Kumamoto Earthquake Disaster Museum Database

Abstract: In order to manage the data on the Kumamoto earthquake that occurred in 2016 (Heisei 28), we created a database system called “Kumamoto Earthquake Disaster Museum DB” in our laboratory. In this DB, we plan to register diverse and large amount of information in the future, and it is necessary to systematize data for data usage and pay attention to how to retrieve efficient data. In this paper, for the purpose of improving the accuracy of “search tag” function, experiments for automatic generation of tags utilizing image recognition processing by Deep Learning, and experiments for occurring in models generated from characteristic learning data differences in detection accuracy were compared from the viewpoints of recall, conformity, and accuracy.

Keywords: Geographic database, Spatial data management, Multimedia database

1. はじめに

2016年(平成28年)4月14日に,熊本県を中心に発生した熊本地震による道路や周辺地域の被害をデータに残して管理するために,本研究室に熊本震災ミュージアムDB(以下熊本震災DBと呼ぶ)を構築し,関係者に公開している.この熊本震災DBには,画像を中心にあらゆるデータを保存し,1次データの管理元として利用している.

熊本震災DBは,データ登録作業時に写真に写っている

震災事象をタグ情報として付与することにより,閲覧時に検索の精度を向上させることができる.しかし,多量のデータを人手で登録すると,震災事象を抽出する際の判断や精度の一貫性を確保することが難しく,ひいては検索の精度が低下するという問題が発生した.

そこで,深層学習の技術を用いることにより,写真から震災事象抽出の判断・精度の一貫性を確保することで,検索の精度を向上させる研究を進めているので紹介する.本研究では,フレームワークとしてDarknetを使用し,アルゴリズムはYOLOを使用した.

¹ 情報処理学会
IPSI, Chiyoda, Tokyo 101-0062, Japan
^{†1} 現在,福岡大学
Presently with Fukuoka University

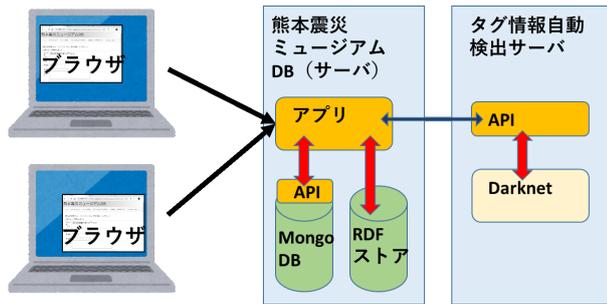


図 1 熊本震災ミュージアム DB の構成

Fig. 1 Composition of Kumamoto Earthquake Disaster Museum DB

2. システム構成

本システム全体の構成は図 1 の通りである。

ユーザはブラウザから本システムにアクセスする。本システムの内部では、MongoDB と RDF ストア (Virtuoso) を利用している [1]。本研究で使用する Darknet は、処理負荷が高いこと、また、GPU を有する環境が望まれることにより、タグ情報自動検出サーバとして熊本震災 DB とは独立したサーバとした [2]。

3. 画像認識フレームワーク「Darknet」

Darknet とは、C 言語と CUDA で書かれた、オープンソースのフレームワークである [3]。Darknet は「学習」を行うことで、任意に指定した物体を認識させることが可能である。

「学習」には、訓練データの作成が必要である。訓練データとは、(1)「学習」に使用する学習させたい物体が写っている画像、(2) その画像の中で検出したい物体を矩形で範囲指定した時の、矩形の座標情報が記述されたテキスト形式のアノテーションファイル、以上の 1 セットを指す。これを複数セット用意し Darknet に学習させると、認識させたい物体を学習したモデルが作成される [4][5]。

Darknet には、標準のモデルも併せて公開されているが、「地割れ」「家屋倒壊」などの震災事象は登録されていない。そこで、「地割れ」を認識するモデルを作成することとした。深層学習による画像認識機能は、形あるものを認識させることが一般的であり、「不定形の事象」を認識できるかは不明である。そこで本研究では、そのような事象を認識できることの確認と、さらに認識の精度を上げるための実験を行った。

4. 実験

4.1 実験概要

「熊本震災で発生した「震災事象」を写した訓練画像」を用いて作成したモデルと、「インターネット上でダウンロードした様々な場面で発生した「震災事象」を写した訓

表 1 画像の特徴
Table 1 Features of images

	熊本震災の画像	インターネット上の画像
撮影の時間帯	日中	日中
撮影時の気候	雨天無し	雨天無し
撮影対象	熊本震災	東日本大震災・阪神淡路大震災など様々
撮影場所	熊本県	東北地方・近畿地方など
画素数	ほぼ均一で高画質	高画質・低画質共に有り
撮影に使用したカメラ	同一	異なっている可能性大

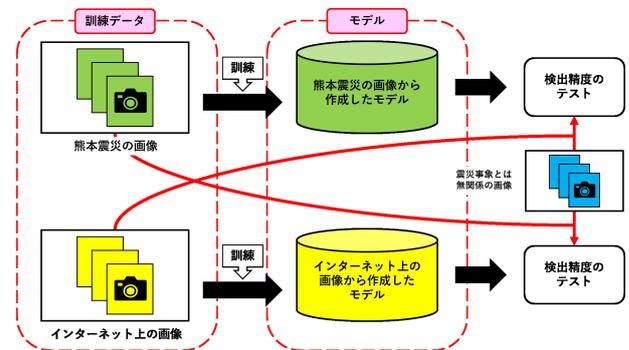


図 2 実験の流れ

Fig. 2 Flow of experiment

練画像」を用いて作成したモデルの 2 つの検出精度を比較して、どのような違いが出るのかを実験する。熊本震災の画像と、インターネット上の画像それぞれの特徴は表 1 の通りである。

なお、本研究では、実験の手法と結果を簡潔かつ明瞭にするため、検出対象を「地割れ」に絞って実験を行った。

次に、「熊本震災の地割れ画像」と「インターネット上の地割れ画像」のそれぞれを別々に Darknet に訓練させていき、モデルを 2 つ作成する。

また、新しく「地割れ」の写っていない「震災とは無関係の画像」を用意する。これは、「背景 (= 「地割れ」でないもの)」を「地割れ」として誤検出する可能性を考慮するためである。

そして、計 3 つの画像群を用意した後に、「熊本震災の画像で作成したモデル」には、「インターネット上の画像」と「地割れの写っていない画像」の 2 つを用いて検出の有無及び精度のテストを行い、「インターネット上の画像で作成したモデル」には、「熊本震災の画像」と「地割れの写っていない画像」を用いて検出の有無及び精度のテストを行う。以上の一連の流れを、図 2 に示す。

その後、それぞれのモデルを用いて検出を行なった時の、(1) TP : 「地割れ」を「地割れ」として検出した画像の

表 2 カウントする項目の関係

Table 2 Relationship of items to count

-	検出有り	検出無し
地割れ有り	TP	TN
地割れ無し	FN	FP

枚数

- (2) TN : 「地割れ」を「背景」として検出した画像の枚数
- (3) FN : 「背景」を「地割れ」として検出した画像の枚数
- (4) FP : 「背景」を「背景」として検出した画像の枚数をそれぞれカウントする. カウントする項目の関係は, 表 2 の通りである.

最終的に, カウントした 4 つの値から, 再現率・適合率・正解率を計算して, 精度の評価を行う.

$$\text{再現率} = \frac{TP}{TP + TN}$$

$$\text{適合率} = \frac{TP}{TP + FN}$$

$$\text{正解率} = \frac{TP + FP}{TP + TN + FN + FP}$$

4.2 実験の詳細

本実験に使う画像群として, 以下のものを用意した.

- (1) 画像群 A : 「実際に熊本震災の影響で起こった地割れ」が写っている画像 107 枚
- (2) 画像群 B : インターネット上でダウンロードした「様々な場面で起こった地割れ」が写っている画像 198 枚

また, 作成したモデルの誤検出の可能性を調査するために, 「地割れ」の写っていない画像群も用意した.

- (3) 画像群 C : インターネット上で無作為にダウンロードした「地割れ」が写っていない画像 46 枚

そして, 画像群 A 及び画像群 B を用いて, 以下のモデルを作成した.

(A) モデル A' : 画像群 A から作成したモデル

(B) モデル B' : 画像群 B から作成したモデル

モデル A' には, 画像群 B と画像群 C を合わせた, 計 244 枚の画像を用いて検出精度のテストを行い, モデル B' には, 画像群 A と画像群 C を合わせた, 計 153 枚の画像を用いて検出精度のテストを行った.

5. 評価

5.1 評価方法

算出された再現率・適合率・正解率の値から, モデルの精度についての評価を行う. 各項目の詳細は以下の通りである.

再現率 : 「地割れ」が写っている全ての画像」の中で, 実際に「地割れ」を検出した確率

適合率 : 「地割れ」と検出した中で, 実際に「地割れ」で

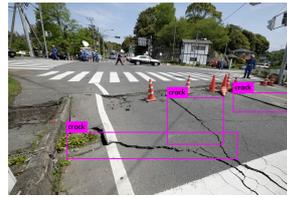


図 3 出力結果 : 1

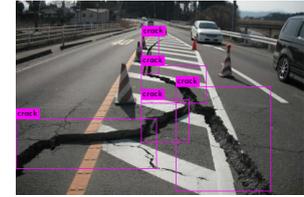


図 4 出力結果 : 2

Fig. 3 Output result : 1

Fig. 4 Output result : 2

表 3 実験結果

Table 3 Experimental result

-	モデル A'	モデル B'
TP 画像数	103	81
FN 画像数	2	11
TN 画像数	95	24
FP 画像数	44	37
再現率	52.02%	77.14%
適合率	98.10%	88.04%
正解率	60.25%	77.12%

あった確率

正解率 : 全体の中で, 正しく「地割れ」と検出される確率

5.2 実験結果

図 3 と図 4 に作成したモデルを用いて熊本震災の地割れ画像に画像認識を使用した際の出力結果と, 表 3 にモデル A' とモデル B' を用いた検出の精度についての結果を示す.

5.3 考察

本研究で行った実験から, 以下の内容について考察した.

- 「不定形の事象の検出」
両モデルの正解率は, 約 60.25%, 77.12% 程度であるが, 図 3・図 4 を見ると, 「地割れ」を認識している部分が見受けられるため, 形状や範囲を特定することが比較的困難なものについても, 画像認識として Darknet が利用可能だというメドが立った.
- 「訓練データの特性によるモデルの精度の違い」
両モデルの正解率を比較すると, モデル B' の方がモデル A' よりも高い値を得ている. 第 4 章 4.1 節の表の通り, 両モデルの元となっている画像群 A と画像群 B との間には, 画素数, 使用したカメラなどに違いがある. 画像群 A は 1 台のカメラで撮影したものであり, 画素数は同一である. 対して画像群 B は, その要素全てが異なっているものである. 一方で, 双方の訓練データの「地割れ」の形状・様子に大きな差異は認められなかったため, 正解率に差異が発生した要因は訓練データのバリエーションであると考えられる.

精度の高いモデルを作成する場合には, 画像の画素数や撮影対象のバリエーションを多くすることが重要だと考える.

6. 結論

本研究では、熊本震災 DB に登録する画像に付与する「検索タグ」の付与の判断・精度の一貫性を確保することを目的に、Deep Learning による画像認識処理の精度を向上させる実験を行った。

熊本震災に関する「地割れ」の検出に成功し、「不定形の事象」の検出にも画像認識が活用できるメドが立った。

「学習データの特性によるモデルの精度の変化」については、訓練データに使用される画像の画素数や撮影対象のバリエーションを増やすと精度の高いモデルが作成されることが判明した。

謝辞 本研究を行うにあたり、多くのご指導を賜りました、福岡大学工学部電子情報工学科画像メディア研究室の鶴田直之教授、廣重法道助教、前田佐嘉志助教、サーバ環境をご提供頂いた福岡大学情報基盤センターの奥村勝教授、様々な意見を頂いた画像メディア研究室の皆様、その他、ご協力頂きました関係者の方々に深くお礼申し上げます。

参考文献

- [1] 舛谷拓也：熊本震災ミュージアムデータベースと阿蘇ジオパークデータベースシステムの構築と利便性に向けた改良，(2018.02).
- [2] 廣重法道、鶴田直之：深層学習を用いた災害関連キーワードの画像への自動ラベル付け機能の開発，(2018.12).
- [3] Joseph Redmon: Darknet: Open Source Neural Networks in C, 入手先 (<http://pjreddie.com/darknet/>) (2013–2016).
- [4] chimamedia: ご注文は YOLOv3 ですか!?! ~ Windows で YOLOv3 を動かす ~, 入手先 (<https://qiita.com/chimamedia/items/c261a858e1c7b8160af1>) (2018).
- [5] demu: YOLO V3: オリジナルデータの学習, 入手先 (<https://demura.net/misc/14458.html>) (2018).