

特集号招待論文

# 深層学習によるコンクリート護岸劣化領域検出システムの開発

齋藤 彰儀<sup>1</sup> 上総 虎智<sup>1</sup> 平木 悠太<sup>1</sup> 天方 匡純<sup>2</sup> 吉田 武司<sup>2</sup>



<sup>1</sup> (株) ブレインパッド <sup>2</sup> 八千代エンジニアリング (株)

従来の河川コンクリート護岸の維持管理業務は、河川技術者の目視による定性的判断により行われており、多大な作業負担となっている。技術者の作業負荷を低減し、生産性向上を図るための方法として、定量的な判断にもとづく劣化判定と検査履歴管理が考えられる。これらをシステムに実装するためには、コンクリート護岸の劣化領域を自動で検出する必要があり、さらに履歴情報や学習データ等のデータを効率的に管理する基盤が必要となる。筆者らは、深層学習技術を用いてコンクリート護岸劣化検出モデルを開発し、クラウド上にWebアプリケーションシステムを構築した。システムの導入期待効果と、深層学習技術を用いた検出モデルの開発からアプリケーション化・運用までの過程において得られた知見をまとめる。

## 1. 河川管理業務における課題

日本国内には数多の河川があり、その附帯施設として洪水を円滑に流す重要な役割を担うコンクリート護岸が設置されている。近年、これらの施設が老朽化の兆しを見せており、維持管理が大きな課題となっている。これまでは、河川技術者（土木技術者）による目視情報を主体とする劣化状況の把握を行っており、定性的な管理となっている。また、設置された時期や地域などによって整備形式が異なり、点検・改修には熟練した技術が必要となる。その結果、河川管理にかかわるコストが膨大になることと、劣化状況の判断基準が技術者個人によってさまざまに異なることが問題となっている。

これらの問題を解決し、河川維持管理業務の生産性を向上させるためには、定量的な判断にもとづく劣化判定と検査履歴管理を行う必要があると筆者らは考えた。そして、これらをシステムとして実現するためには、コンクリート護岸の劣化領域を自動で検出する必要があり、さらに検査履歴情報や学習データ等のデータを効率的に管理する基盤が必要となる[1],[2]。

筆者らは、コンクリート護岸の劣化状況のひとつであるひび割れ（目地の開き）を対象に、深層学習技術を用いた検出モデルを開発し、履歴管理機能とともにクラウド上にWebアプリケーションシステム（GoganGo）を構築した。GoganGoの入力は、デジタルカメラで撮影した（複数の）護岸画像である。護岸画像からひび領域を検出し、入力画像にひびを重ね合わせた画像を作成する。出力はひびを重ね合わせた画像と評価指標である。また、検出実行履歴を管理することができる（, ）。

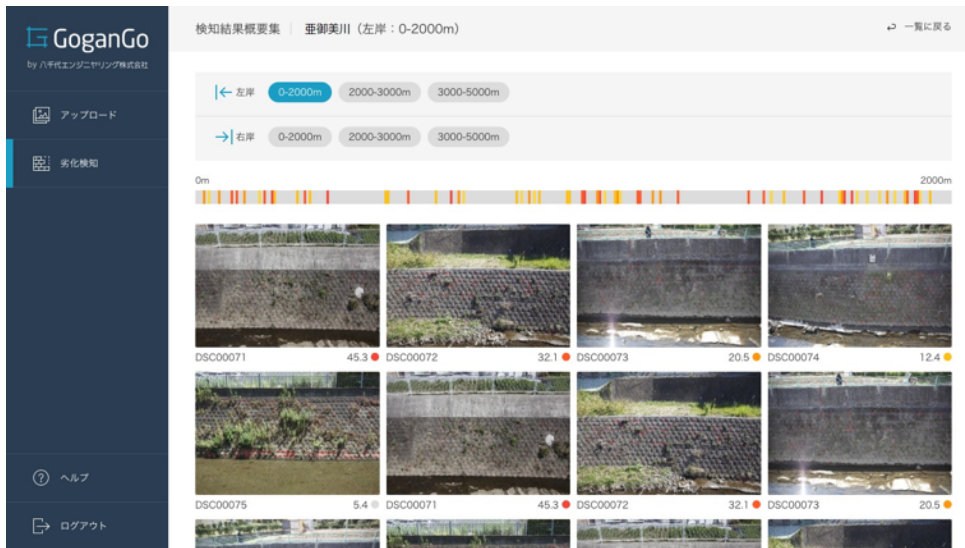


図1 GoganGo河川管理画面イメージ

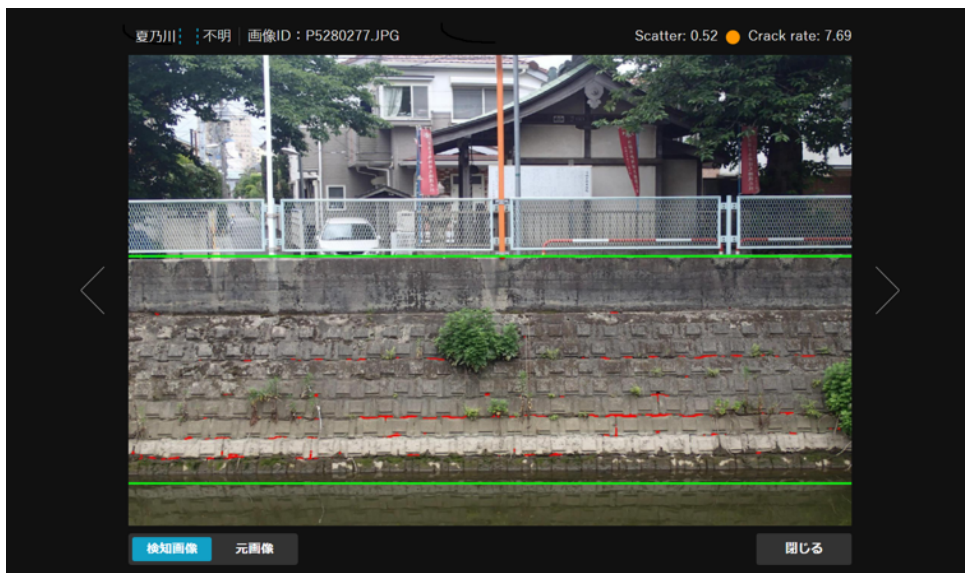


図2 劣化詳細確認画面イメージ

本稿では、システムの概要と深層学習技術を用いた劣化検出モデルの開発からアプリケーション化・運用までの過程において得られた知見を述べる。

本稿の構成は以下となる。第2章では、GoganGoの深層学習モデルについて述べる。ここでは、実際に用いているモデルや、開発プロセスについて述べる。続く第3章では、深層学習システム開発について述べる。筆者らはクラウド上にWebアプリケーションとして構築した。その際のアーキテクチャや開発プロセスについて述べる。第4章では、筆者らが期待しているシステム化による導入効果について述べる。第5章では、深層学習システム開発を通して得られた知見をまとめる。最後にまとめとして、今後の課題とこれからの取り組みについて述べる。

## 2. 深層学習を用いたコンクリート護岸劣化検出モデル開発

GoganGoにはふたつの深層学習モデルが実装されている。ひとつは護岸領域検出モデルであり、もうひとつはひび領域検出モデルである（図3）。どちらも教師あり学習である。開発言語はPythonであり、使用した深層学習のフレームワークはTensorFlow[3]、Keras[4]である。護岸領域検出モデルは、デジタルカメラで撮影した護岸画像から検査管理対象である護岸領域を検出するモデル（物体検出）である。一般に護岸画像には、コンクリート護岸だけでなく、画像上部には建築物や雑木林、下部には河川が写っている。したがって、解析対象であるコンクリート護岸領域をまず抽出する必要がある。

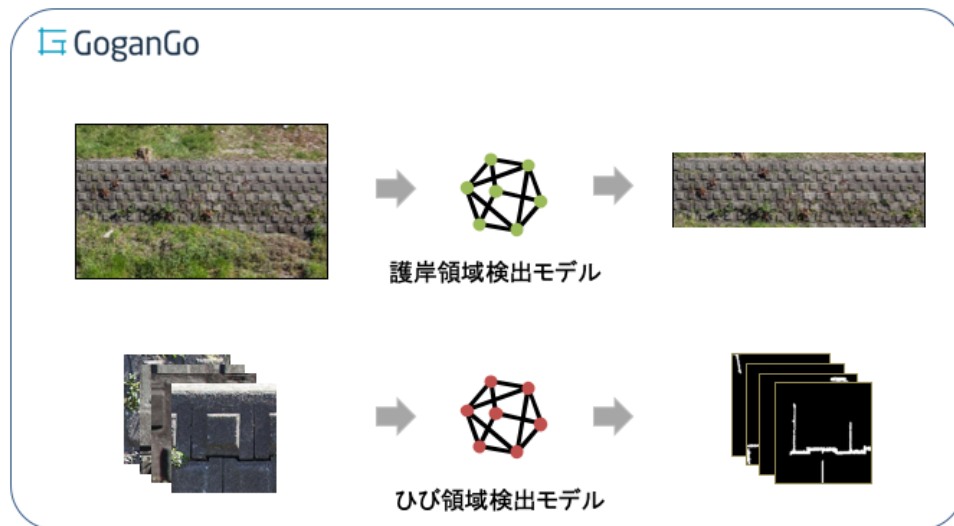


図3 劣化領域検出モデル概要

ひび領域検出モデルは、コンクリート護岸画像からひび領域を検出するモデルである。これは画像をピクセル単位で正常かひびかを分類するモデル（セマンティック・セグメンテーション）である。以下それぞれの詳細を述べる。

## 2.1 護岸領域検出モデル

デジタルカメラにより撮影された画像には護岸が写っており、そこから護岸領域のみを抽出することは、現在の深層学習技術を用いればそれほど困難ではない。しかし、護岸領域検出に失敗すると、続いて実行されるひび領域検出における検出精度や、今後の運用に影響を及ぼす。したがって、写っている護岸領域を確実に検出する必要がある（図3上図）。

ベースとなる検出モデルのフレームワーク（メタアーキテクチャ）は、精度を重視し、Faster-RCNN[5]とした[6]。Faster-RCNNは大きく分けると特徴量抽出器（図4のconv layers部分）と検出器（図4のRegion Proposal NetworkとRoI pooling）からなるモデルである。特徴量抽出器としては、こちらも精度を重視しInception-ResNet-V2[7]を用いた。護岸画像約300枚から護岸領域を記した教師データを作成し、COCOデータセット[8]で事前学習したモデルから、特徴量抽出器は固定し、検出器部分のみを転移学習した[6]。評価用データセットにおける、文献[5]にもとづく分類および回帰損失はともに0.1以下となった。検出結果の例を図5に示す。図中の緑線枠が検出された護岸領域となる。

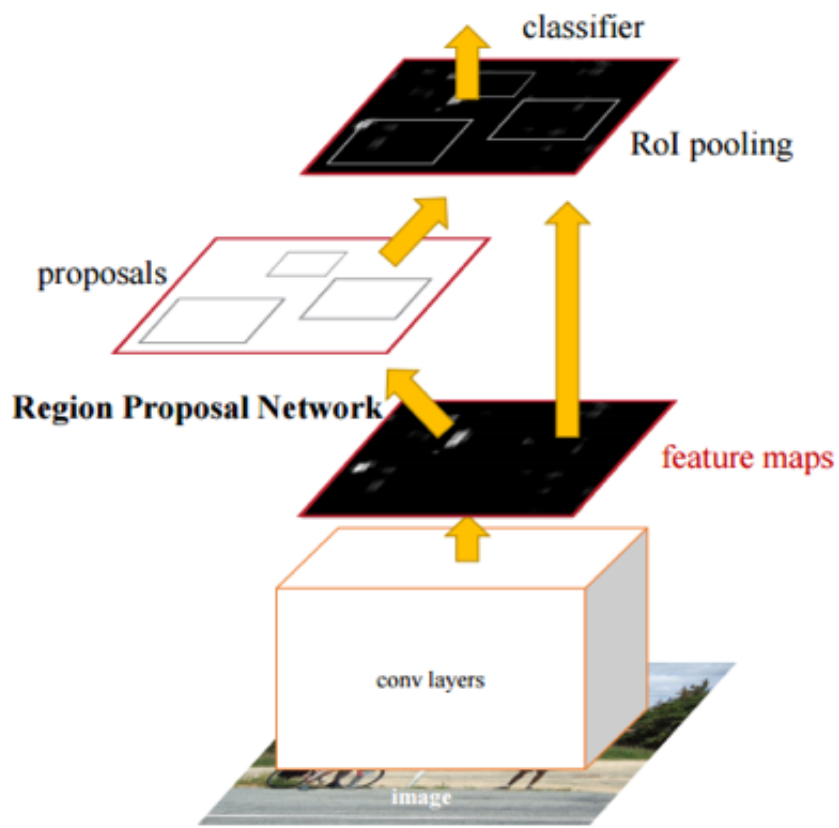


図4 Faster-RCNN概要 (文献[5]より引用)



図5 護岸領域検出結果

## 2.2 ひび領域検出モデル

デジタルカメラによるコンクリート護岸画像と、劣化領域を塗りつぶした画像をもとに、両者の対応関係を学習するモデルをConvolutional AutoEncoder (Encoder-Decoder Model) として構築した (図3下図)。

教師用データは、デジタルカメラで撮影した護岸画像に対し、河川技術者がひびと判断した個所を図6のように赤色で塗りつぶして作成した。これとオリジナルの護岸画像との差分を教師とした。このような画像を約100枚用意し、学習用データセットとした。



図6 教師用データ

ひび領域検出モデル構築の際に参考にしたモデルは、医療画像分析用に開発されたFusionNet[9]である。FusionNetは比較的シンプルな構造であり、後処理を必要とせずEnd-to-Endで学習できるモデルである。具体的にはU-Net[10]とResNet[11]のハイブリッドモデルと言える。このFusionNetを護岸画像の学習用データセットにあわせてモデル構造やハイパーパラメータのチューニングを行った。このときの定量的なターゲット指標は、F値を用いた。ただし、教師用データの作成が人手である点や、画角によりひびの写り方が異なるといった点を考慮し、参考情報程度とし、目視による定性評価も用いることにした。試行錯誤やチューニングの結果、オリジナルのFusionNetからの変更点は以下となった。

- 入力画像サイズの変更  
計算量の観点から入力画像サイズを640×640から224×224へ変更した。これにより、1枚の護岸画像から、224×224クロップ画像が約150枚得られた。
- プーリング層の追加  
入力サイズを調整した結果、7×7サイズまでプーリングが可能となり、4層から1層追加し5層とした。
- 活性化層とバッチ正規化層の位置変更  
畳み込み層の後に活性化層、バッチ正規化層としていたが、この順序を入れ替えた。
- ロングスキップコネクションの結合層  
加算結合 (残差結合) から、U-Netと同様な連結結合へ変更した。この結果、劣化領域の

エッジをよく捉えるようになった。

これらのハイパーパラメータチューニングを行い、構築したモデルは図7となった。図中の左の値はフィーチャマップサイズである。学習における交差エントロピー損失は0.01以下となり、評価用データセットにおけるF値は約0.6となった。検出結果の例は図8のようになり、河川技術者と遜色ないレベルでの検出精度が得られた。

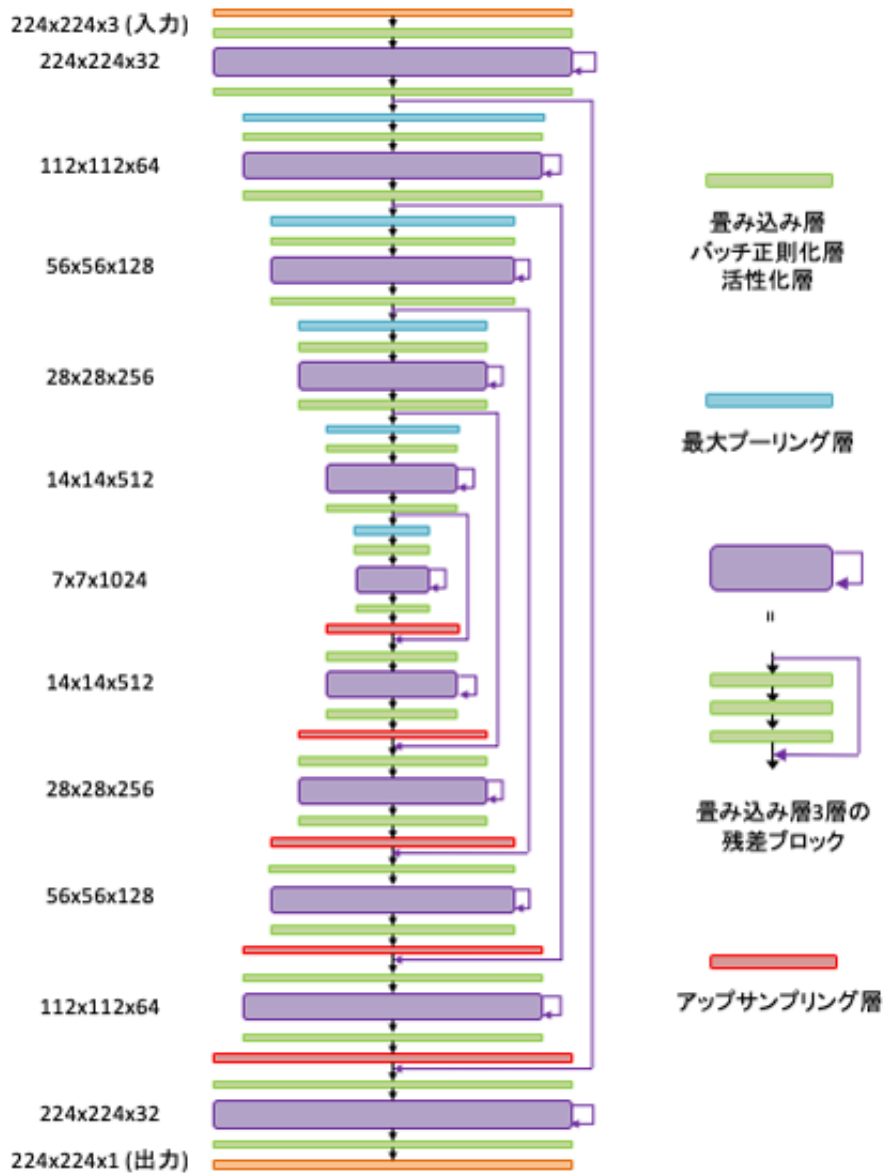


図7 ひび領域検出モデル

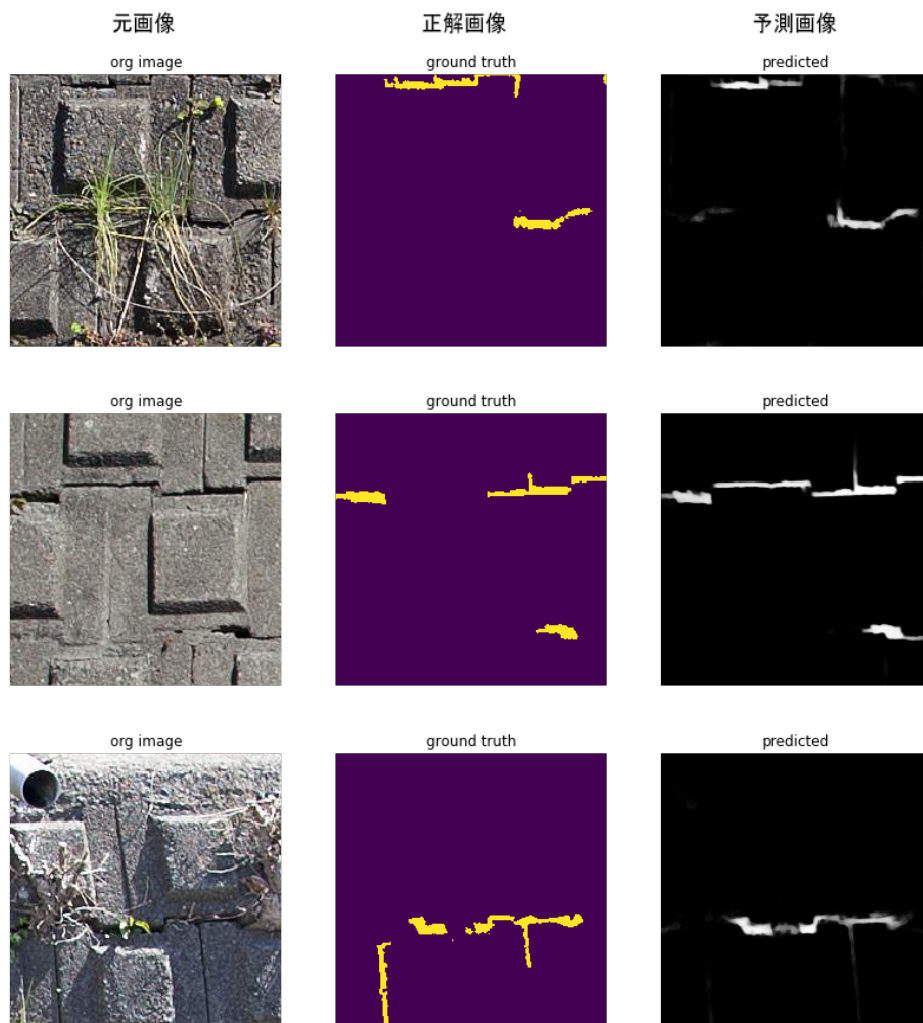


図8 単一（正方形）テクスチャでのひび領域検出結果

### 2.3 深層学習モデル開発・改善プロセス

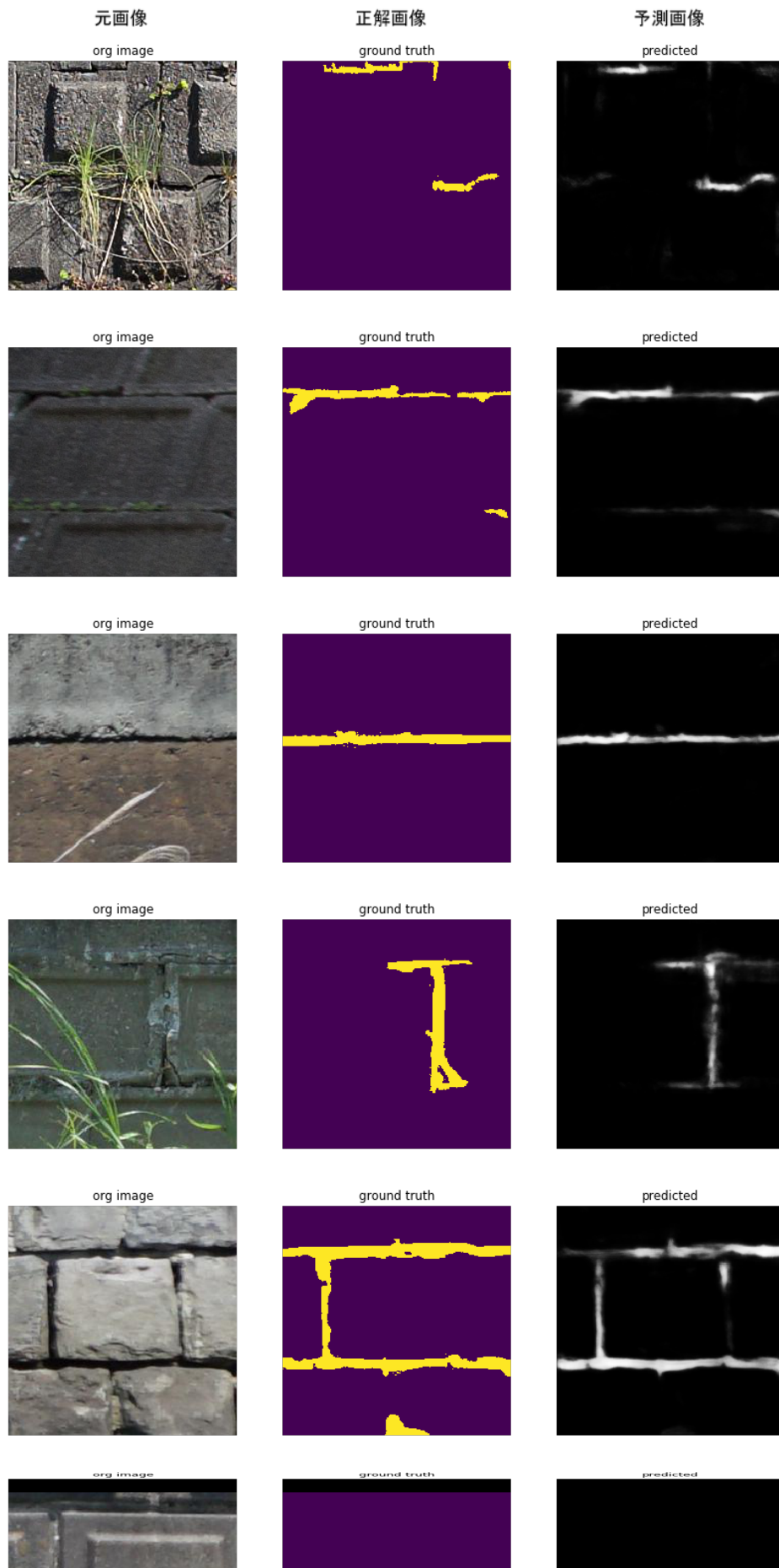
機械学習（深層学習）モデルの開発スタイルは、従来のアルゴリズム・プログラム開発とは異なる。従来は、プログラムコードを書いてシステムの動作を定義していった。しかし機械学習モデルは、データから帰納的に定義される。したがって機械学習モデルの開発では、これまでの開発方法論や品質保証技術等が必ずしも適用できず、技術的な負債が発生しがちとなる[12],[13],[14]。

こういった課題に対しては、まずは概念検証（PoC）を行い、実現可能性等を検討する。その後、PoCの結果を受け、次の開発や施策につなげていくといったプロセスを採ることが多い。GoganGo開発においても、まずは正方形型の単一護岸テクスチャ（コンクリート表面模様）に対して、PoCとしてひび検出の技術検証を行った（図8）。

一般に河川には複数の護岸テクスチャが整備されており、ひとつの河川においても数種類の護岸テクスチャが存在する。単一のコンクリート護岸テクスチャを学習したモデルでは、その他のテクスチャにおけるひびを十分に検出できない。そこで、PoCの次施策として、複数種類の護岸テクスチャにおけるひび検出の技術検証を行った。新たな学習用データの作成は、クラウドソーシングを用いた。こうして得られた数種類の護岸テクスチャ画像データは合計で約500枚となっ



た. ここからテクスチャごとのデータ量を調整して学習用データセットとした. これらのデータセットにおいてひびを改めて学習させたところ, 評価用データの各テクスチャにおいて, 適切なひび検出精度が得られた (図9).



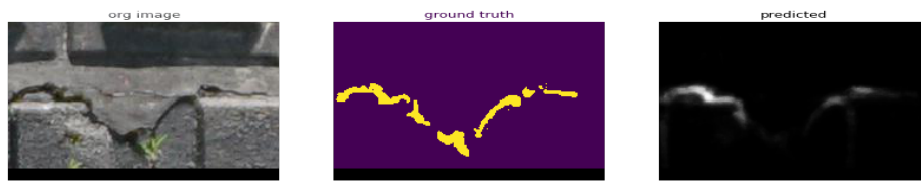


図9 複数テクスチャでのひび領域検出結果

### 3. クラウド上へのアプリケーション構築

河川技術者の生産性を向上させるためには、深層学習モデル単体では不十分であり、実業務を意識したアプリケーション化が必要不可欠である。

深層学習技術全般の発展により、深層学習モデルをアプリケーションに組み込むことで、従来システムでは実現が困難であったことが実現可能となってきている。それに伴い、ユーザの深層学習システムに対する期待値が大きくなることもある。また、業務利用のイメージが湧きにくいといった課題も発生する。このような理由により、深層学習システムのユースケースや要件が不明確となりがちとなる。したがって、GoganGo開発ではプロトタイピング開発手法を採った。限定的な機能を実現したプロトタイプを作成し、開発の初期段階から完成品のイメージをユーザと共有でき、ユーザからのフィードバックが得られるので、効率的に開発が進められる。

プロトタイプの構築には、GPUやストレージ等のリソースや豊富なマネージドサービス、今後の運用や拡張等を考慮し、クラウド上（Google Cloud Platform）にWebアプリケーションとして構築することにした。使用した言語は、深層学習モデル開発と同じくPythonを用いた。

プロトタイプにて開発した機能は、護岸画像ファイルのアップロード、劣化検出実行設定、劣化検出実行および確認、検出履歴管理、ダウンロードといった基本的な機能とした。アーキテクチャを図10に示す。

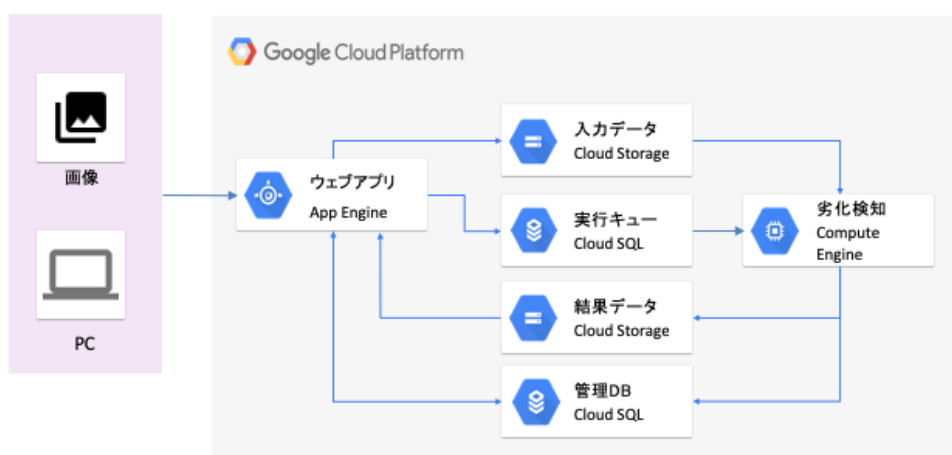


図10 プロトタイプアーキテクチャ

Webフロント側のアプリケーションは、フルマネージドプラットフォーム上に構築した。劣化検出機能のバックエンドはGPUを1枚搭載したLinux VM上に構築した。管理等のDBはマネージドDBサービスを利用した。

クラウドサービスおよびフレームワークを効率的に利用し、基本機能のみとしたことにより、開発期間は6人月程度となった。

## 4. 検証事例および期待される効果

### 4.1 検出結果の検証事例

前年度の点検記録結果を踏まえ、GonganGoのひび検出能力を実業務の中で検証した。ただし対象とした護岸は、健全度（4段階評価）のうち、最も悪い状態とその次に悪い状態の護岸を対象とした。理由は技術者の判断が比較的安定していると考えられるからである。

これらの護岸画像から、GonganGoがひびを検出できるかを検証した。その結果、撮影条件が悪くGonganGoの入力としては不適切な画像も含まれていたが、それらを含めて7割以上を検出することができた。

今後の検出力向上には、学習用データのバリエーションを増やすことを考えている。また、未知のテクスチャに対応するための再学習機能の実現も考えている。

### 4.2 期待される効果

現在の維持管理を目的とした河川コンクリート護岸の状況確認は、図11の上段フローに従い実施されている。現状フローは、

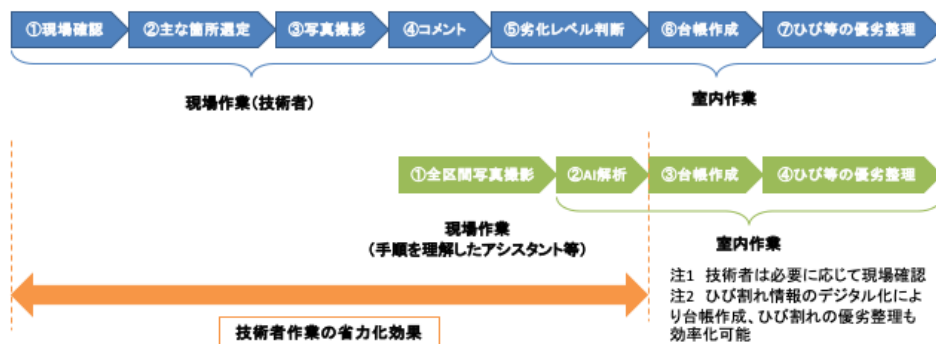


図11 AI導入前後の維持管理行為の差異

- ① 全体（あるいは部分）区間のコンクリート護岸のひびの状況・位置の概要確認
- ② 主なひびの箇所（あるいは区間）の選定
- ③ 選定箇所・区間のひびの写真撮影
- ④ ひびの状況コメント（文字記録）
- ⑤ ひびの劣化レベル判断
- ⑥ 台帳作成
- ⑦ ひびの状況の優劣整理

等に細分化される。このうち、GonganGoを活用してひびの自動検出が可能になると、河川技術者による上記①から⑤のフローが、アシスタント等による全区間の写真撮影、GonganGoによるひび検出といったフローに代替される。この結果、河川の維持管理に携わることができる

人々の裾野が広がり、技術者不足の解消や生産性向上に繋がる。また、GoganGoによるひび検出結果はデジタル情報であるため、台帳作成の労力軽減も可能となり、経年的な変化追跡も定量的に可能となる[1],[2].

今後は導入事例を増やし、どのくらい生産性が改善しているかを検証する予定である。

---

## 5. 深層学習システム開発を通して得られた知見

---

ここでは、GoganGo開発を通して得られた知見をまとめる。深層学習モデルパート、システム開発パートそれぞれに分けて述べる。また応用可能性についての考察を与える。

### 5.1 深層学習モデル開発において得られた知見

#### 5.1.1 ターゲット指標の設定

第2章にも述べた通り、深層学習モデルはデータから帰納的に定義される。したがって、推論結果における100%の精度指標の達成は期待できない。そこで、運用や将来を見越した現実的な精度指標の設定が必要となる。

護岸メンテナンスにおいては、誤検出は多くとも、ひびの検出漏れを減らしたいとの要望があったため、まずは再現率向上をターゲットにチューニングを行った。しかし、モデル開発が進むにつれ、モデルのひびに対する検出感度が強くなり過ぎた。そこで、目視による定性評価と親和性の高い指標であるF値へとターゲット指標を変更した。その結果、河川技術者と同程度の検出精度を得ることができた。

#### 5.1.2 管理指標の開発

ひび領域検出という限定的な機能であるにもかかわらず、河川管理・台帳用のメトリクス（管理項目）開発の視野が広がった。従来は、河川管理者がひびの写真とともに大きさや場所等といった管理指標を独自フォーマットで記録していた。それが深層学習モデルによる統一的な判断が可能となり、従来では獲得困難であった管理指標の検討が可能となった。生産性向上を目指して開発が行われたが、新たな付加価値を生む結果となった。

GoganGoでは、護岸領域におけるひびの割合と、ひびの分散度合を指標とした。ひびの割合は、護岸領域におけるひびピクセルの総数を護岸領域ピクセル総数で割ったものである。分散度合いは、224×224サイズのクロップ画像において、ひびの割合が0.5%以上のクロップ画像の数を足したものである。

また、これら以外にも実運用や他の劣化状況を踏まえた指標を検討している。

#### 5.1.3 学習用データの品質

高精度な深層学習モデルを作成するには、大量の高品質な学習用データの準備が不可欠となる。モデルの精度は、学習用データの品質に依存する。品質にばらつきがあると、精度向上は期待できない[15].

ひび領域検出モデルの開発では、PoCの段階で約2100万画素の護岸画像を約100枚用意した。これから護岸領域を224×224サイズにクロップした結果、モデル開発に十分な量が得られた。また実験の結果、224×224サイズの画像約5,000枚でF値の上昇が収まってきた。

対応テキスト拡張にあたりクラウドソーシングを用いて学習用データを作成した。同一護岸画像に対して、複数のクラウドワーカが作成した学習用データを統計処理することにより、河川技術者によるものと同程度の品質を確保することができた。実際に、単一のクラウドワーカ作成による学習用データのみで学習しても、F値が期待する値までの向上はみられなかったが、統計処理を実施した学習用データで学習するとF値が向上した。短期で高品質、低コストな学習用データを得ることができた。

## 5.2 深層学習モデルを含んだシステム開発において得られた知見

### 5.2.1 大規模データの扱い

膨大な量の画像データ（検出前および検出後のデータ）を管理するために、分散オブジェクトストレージと呼ばれるスケラブルなストレージを使用した。GCP上ではGoogle Cloud Storage (GCS)として提供されている。GCSはほぼ無制限にスケールさせることができるので、保存容量を意識することなく開発を進めることができた。また、同一リージョン内であれば高速に読み書きができる。

同様に、深層学習モデルもファイルサイズが大きいため、GCS上で管理した。具体的にはGCSのモデルへのURLを設定ファイルに記載する形で管理した。複数バージョンを保持することができ、バージョンの変更が必要になった際には迅速に対応することができた。また、前バージョンへのフォールバックも容易であった。

一方で、大容量な画像ファイルを恒久的に保存することは、コスト的に非現実的である。パージ計画を検討し、機能として実現する必要がある。ただし、学習用データの管理は重要であり、慎重に設計する必要がある。

### 5.2.2 クラウドアーキテクチャ

深層学習機能は単一VM上に構築しており、バッチ処理として護岸領域および劣化領域検出を行っている。このバッチ処理は一般的な河川全体を処理した場合、各種検出に要する時間は数十時間に及ぶ。ユーザ数が増えた場合、複数VMを並列実行する等のスケールアップ対応が必要となる。GoganGoの場合、検出実行の需要が一定ではない可能性があるため、単純にVMの数を増やす対応では使用率の低いVMが増える可能性があり、コストに見合わない。そのため動作中のVMを効率的に使用する仕組みが必要となる。オートスケール機能や別のマネージドサービス（Cloud ML Engine）の利用を検討している。

### 5.2.3 要件・ユースケースの確認

プロトタイプを開発したことにより、今後の機能開発の方向性や運用方針が可視化されてきた。まず第一に、地図連携機能と管理台帳機能の実現が考えられる。実際の護岸メンテナンス工事は、数十メートルから数キロといった単位で施工される。このような実務に合わせたスケールでの表示や管理指標を開発する必要がある。

また、今後ビジネス展開するにあたり、ユーザ管理や課金管理、ログ管理等の機能を支える基盤要素技術の開発も必要となる。

## 5.3 応用可能性についての考察

GoganGoにおける検出モデルの開発アプローチや深層学習モデルは、他のコンクリート構造物の劣化検出においても応用可能と思われる。実際に問合せもある。

モデル部分に関しては、コンクリート護岸の劣化検出以外においても、作成した工業製品等の品質検査（傷の有無や変色等）への応用が可能と考えられる。

一方で、アプリケーション部分（コンクリート護岸メンテナンス）においては、同じコンクリート構造物でも対象領域によっては法律的な関係もあり、直接的な応用は難しいと考えられる。場合によっては、業務運用方針や既存システム、データの活用方針等との関係も考慮していく必要がある。

---

## 6. まとめと今後の課題

---

本稿では、深層学習システムのプラクティス事例としてGoganGoの開発事例を紹介した。GoganGoの開発目的・課題は、河川技術者が今後不足し維持管理業務に支障をきたすため、技術者の生産性を向上させる必要がある点、技術者により点検・評価業務にばらつきがある点、劣化の経年変化を追跡したい（壊れてから改修ではなく、機能停止を予測し機能停止前に改修して被害を抑えたい、維持管理・改修等のメンテナンス全体費用を抑えたい）といった点である。このような明確な目的・課題を設定し、護岸のひび検出という小さなテーマから取り組んだ。

現在もサービス展開に向けて研究・開発中である。具体的には以下の機能開発を予定している。

- 検出対象の拡大
- 再学習機能の実現
- 多様な撮影条件への対応
- アーキテクチャ基盤の開発

コンクリート護岸には、さまざまな劣化が生じる。劣化領域をすべて同じ技術で検出するのではなく、捉える劣化領域の特性により検出技術を変えていく予定である。それに伴い、検出速度といった非機能や、検出精度の確保等も留意していく予定である。また、他のコンクリート構造物への展開も視野に入れている。

検出対象の拡大と同時に再学習機能の実現も検討している。再学習機能の実現には、学習用データセットと深層学習モデルの管理機能が必要となる。精度の高い深層学習モデルを維持・構築するためには、入力される学習用データに対して、必要最低枚数や品質チェックといったいくつかの制約を設ける必要がある。また、これらの制限をユーザに促す必要も生じる。モデル管理においては、学習に利用したデータセットの構成やハイパーパラメータの設定情報、評価指標およびその結果等を管理する必要がある。

GoganGoの入力画像は、晴天時にデジタルカメラで撮影された画像を想定している。しかし運用を考慮すると、河川の状況やカメラの性能により必ずしも想定された撮影状況の画像は得られないと考えている。さまざまな画像データを用意して学習用データセットとする方法や、画像処理によって補正する等の方針が考えられる。

プロトタイプはシンプルな構成であり、変更が生じて柔軟に対応することができる。しかし今後の機能拡張や機械学習モデルの追加等を考慮すると、運用の属人化やコスト上昇といったリスクが考えられる。これらを回避するには、サポートツールの導入やアーキテクチャの定期的な見直し（マネージドサービスを積極的に用いたマイクロサービス化等）が必要であると考えている。幸いにも機械学習に関する技術革新は速いので、積極的に導入したいと考えている。

最後に近々の研究課題としては、実際の点検結果の利用方法に関する研究がある。検出方法（プロセス）の決定や、これを実現するための研究、導入効果、経年劣化追跡技術の研究等である。ユーザが抱く機械学習モデルに関するネガティブな意見に対しても、対処方法を検討する必要がある。また、河積確保に影響する堆積土砂や樹木等を検出することも目標としている。

## 参考文献

- 1) 天方匡純, 吉田武司, 藤井純一郎: 深層学習方式を活用した河川のコンクリート護岸の劣化領域抽出, 第73回年次学術講演会, 土木学会 (2018) .
- 2) 天方匡純: 深層学習方式を活用した河川のコンクリート護岸の劣化領域抽出, 建設機械施工, Vol.70, No.5, (May 2018).
- 3) Tensorflow : <https://www.tensorflow.org/>
- 4) Keras : <https://keras.io/>
- 5) Ren, S., He, K., Girshick, R. and Sun, J. : Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.39, No.6, pp.1137-1149, (June 2017).
- 6) Huang, J. et al. : Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3296-3297 (2017).
- 7) Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. : Inception-v4, Inception-ResNet and The Impact of Residual Connections on Learning, in Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (2017).
- 8) Lin, T-Y. et al. : Microsoft COCO : Common Objects in Context, arXiv : 1405.0312 (2014).
- 9) Quan, T. M., Hildebrand, D. G. C. and Jeong, W-K. : FusionNet : A Deep Fully Residual Convolutional Neural Network for Image Segmentation in Connectomics, arXiv : 1612.05360 (2016).
- 10) Ronneberger, O., Fischer, P. and Brox, T. : U-Net : Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS, Springer, Vol.9351, pp.234-241, (2015).
- 11) He, K., Zhang, X., Ren, S. and Sun, J. : Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770-778 (2016).
- 12) 科学技術振興機構 研究開発戦略センター : AI応用システムの安全性・信頼性を確保する新世代ソフトウェア工学の確立 (2018) .
- 13) Sculley, D. et al. : Machine Learning : The High Interest Credit Card of Technical Debt, SE4ML : Software Engineering for Machine Learning (NIPS 2014 Workshop) , (2014) .
- 14) Sculley, D. et al. : Hidden Technical Debt in Machine Learning Systems, Advances in Neural Information Processing Systems 28 (NIPS2015), pp.2503-2511, (2015) .
- 15) Baylor, D. et al. : TFX : A Tensor Flow-Based Production-Scale Machine Learning Platform, Proceedings of The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1387-1395 (2017).

齋藤 彰儀 (非会員) [akinori.saito@brainpad.co.jp](mailto:akinori.saito@brainpad.co.jp)

(株) ブレインパッド アナリティクスサービス本部.

上総 虎智 (非会員) [taketoshi.kazusa@brainpad.co.jp](mailto:taketoshi.kazusa@brainpad.co.jp)

(株) ブレインパッド アナリティクスサービス本部.

平木 悠太 (非会員) yuta.hiraki@brainpad.co.jp  
(株) ブレインパッド アナリティクスサービス本部.

天方 匡純 (非会員) amakata@yachiyo-eng.co.jp  
八千代エンジニアリング (株) 技術創発研究所.

吉田 武司 (非会員) t-yoshida@yachiyo-eng.co.jp  
八千代エンジニアリング (株) 技術管理本部.

採録決定：2019年1月28日

編集担当：福島 俊一 (科学技術振興機構(JST) )