

生成 AI を活用した GUI 変更検出 GUI Change Detection using Generative AI

和久井 拓[†] 増田 峰義[†]
Taku Wakui Mineyoshi Masuda

1. はじめに

企業の IT 環境の運用、管理、保守を代行するマネージドサービスは近年、DX による IT 利用拡大や IT 人材不足を背景にその需要が高まっている。本稿は、生成 AI を活用したマネージドサービスの運用業務の効率化を目的とし、クラウドサービスの管理コンソールを対象に、Graphical User Interface (GUI) の変更を自動検出する技術を提案する。本技術はクラウドサービスの GUI 変更に伴う作業中断、予定外の手順書更新作業の発生を防止することができる。

2. 背景

2.1 運用作業

マネージドサービスは、サービス提供者が顧客企業の IT 環境の運用、管理、保守を代行することで、企業は自社のリソースをコア業務に集中させることが可能となる。これにより、運用コスト最適化、システム可用性向上、セキュリティリスク低減などの効果が期待される。マネージドサービスで管理する IT 環境は、例えば顧客企業の Amazon Web Service (AWS) 環境が挙げられる。マネージドサービスを利用することで、顧客自身で AWS のマネジメントコンソールや Command Line Interface (CLI) での難しい操作をせずに AWS の IT リソースを利用することができる。サービス提供者側では、作業担当者が、各作業に対応する手順書を参照しながら、管理コンソール上で依頼された作業を依頼された日時に実行する。手順書は、誰が実行しても等しく正確に作業が完了できるように、操作内容が 1 ステップずつ詳細に記載されており、また操作画面のスクリーンショットなども添付されている。

2.2 GUI 変更が運用作業へ与える影響

2.1 節に前述の通り、マネージドサービス業務においては、AWS をはじめとする各種クラウドサービスを操作対象とする場面が少なくない。これらのクラウドサービスは、Web ブラウザ上で視覚的に操作可能な管理コンソールを提供しており、GUI を介した IT リソースの設定や管理が可能である。しかし、これらの管理コンソールは、運用上の改善や新機能の追加、既存機能の廃止といった理由により、その操作方法やインターフェースのデザイン、提供機能の名称などが随時変更される場合がある。

図 1,2 に管理コンソールにおける GUI 変更の事例を示す。この例では、AWS Backup[2]で提供されている機能の 1 つが「バックアップポータル」から「ポータル」に変更され、ページのレイアウトが大きく変更されている。また、ページ下部の表がタブで切り替えられる GUI が追加されている。クラウドサービスのこのような変更は、一般的にユーザビリティの向上やサービス拡充を目的として実施される。

[†]株式会社日立製作所 研究開発グループ
Hitachi, Ltd. Research & Development Group

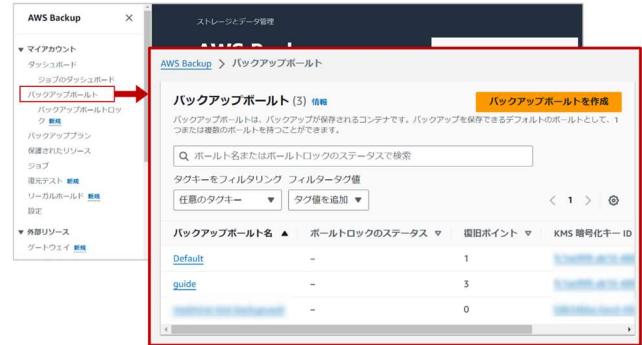


図 1 管理コンソール上の変更事例 (変更前) [1],[2]



図 2 管理コンソール上の変更事例 (変更後) [1],[2]

しかし、マネージドサービスにおいては、これらの変更が運用に影響を及ぼす可能性がある点に留意する必要がある。例えば、マネージドサービスの手順書に掲載されている操作画面のスクリーンショットは、手順書作成時点でのスクリーンショットである。ここで、手順書作成時から手順書作業を実行するまでの間に操作画面上に変更があった場合、手順書のスクリーンショットや説明が、操作中の画面の GUI や表記と異なる事態が発生する。特に、操作に係る箇所に変更があった場合、手順書の指示通りに操作できない可能性がある。マネージドサービスの運用作業において作業担当者は誤操作防止のために、原則として手順書に記載されていない操作を個人の判断で実行することができない。そのため、作業担当者が手順書に記載の操作に関わる GUI 仕様の変更気付いた場合、確認のために中断しなければならない場合がある。作業の中断は顧客の業務にも影響し得る。また、作業担当者からの問い合わせなどで GUI の仕様変更に気付いた手順書作成者は、作業担当者への対応に加え、手順書を更新しなければならない。

以上のように、GUI 変更は作業依頼元、作業担当者、手順書作成者へ影響を及ぼす場合がある。変更箇所が複数の手順書に関係する場合、影響を受ける範囲や手順書作成者の負担はさらに大きくなる。この課題に対し本稿は、管理コンソールのスクリーンショットを定期的に自動取得し、前回取得したスクリーンショットとの差分をマルチモーダル生成 AI によって検出する GUI 変更検出システムを提案する。

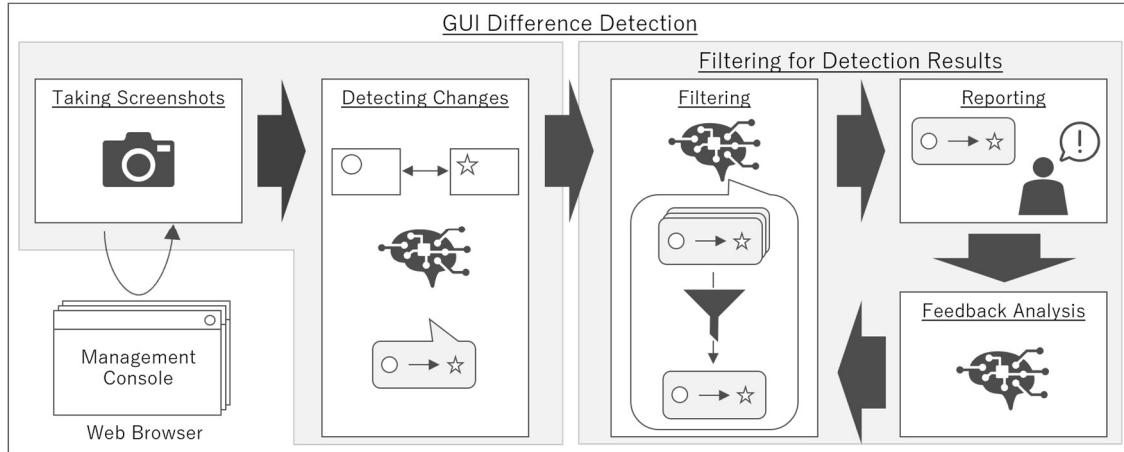


図3 GUI変更検出技術の全体像

3. GUI変更検出技術

本稿で提案するGUI変更検出技術は、画像差分検出と過検出抑制フィルタより構成される。図3にシステムの全体像を示す。画面差分検出では、管理コンソールの各ページを定期的に取得し、過去に取得した同ページのスクリーンショットとの差を生成AIに比較させる。過検出抑制フィルタは、画像差分検出において検出された差分のうち、運用作業に関係しない報告不要な差分を生成AIに除去させ、報告が必要なGUI変更のみを抽出する。

3.1 画像差分検出

画面差分検出は、スクリーンショット取得部と画像比較部より構成され、画像比較部には画像を扱うことができるマルチモーダルモデルを活用する。

まず変更検出の対象とする管理コンソール中のページURLが記録されたURLリストを用意する。検出対象は、例えば手順書にスクリーンショットが掲載されているページである。スクリーンショット取得部では、URLリストに記載されているURLに順番にアクセスし、各ページのスクリーンショットを取得する。スクリーンショットは定期的に取得され、システム内で日付毎に管理される。

画像比較部は、最新のスクリーンショットと、次に新しいスクリーンショットをページごとに比較する。例えば、スクリーンショットを毎日取得する場合、当日のスクリーンショットと前日のスクリーンショットをマルチモーダルモデルに入力し比較させる。このときマルチモーダルモデルの出力は差分の有無と内容である。差分がある場合には、併せてその差分の内容をテキストで出力する。

3.2 過検出抑制フィルタ

画像差分検出では、マルチモーダルモデルがスクリーンショット間の違いを検出する。しかしクラウドサービスの管理コンソール上では、運用作業に影響しない変化が多く発生する。例えば、バナーなどによる通知の有無やリソースの値の変化、日々更新されるニュースなどである。これらは確かに管理コンソール上の変化であるがGUI変更とは関係がなく過検出と言える。過検出は、手順書に基づく運用作業には影響しないため、作業担当者や手順書作成者への報告は不要である。画面上の変化がGUIの仕様に関わるものか否かの判断は、普段から管理コンソールを使用して

いる人には容易であるが、生成AIはその判断基準や感覚を持ち合わせていない。本稿では、報告要否の判断基準をユーザが補強し、その基準を反映して生成AIが過検出を除外する過検出抑制フィルタを開発した。

過検出抑制フィルタは、画像差分検出結果に対するフィードバック、生成AIによるフィルタ条件の生成、生成AIによるフィルタの適用という流れで実現される。まず、ユーザは、画像差分検出結果の中で報告不要な検出結果に対してフィードバックを与える。フィードバックには、報告不要と判断した根拠が含まれる。次にフィードバックを生成AIに分析させる。この分析では、ユーザのフィードバックを生成AIに入力し、1. フィードバックの内容に対するクラスタリング、2. クラスタ毎にフィードバックの内容を集約し清書、3. 清書されたフィードバックの内容をリスト化し出力、という手順を指示する。こうして、フィルタすべき検出結果の特徴を箇条書きでまとめたフィルタ条件を得る。生成されたフィルタ条件は、以降の画像差分検出結果に対するフィルタで適用される。画像差分検出結果とフィルタ条件を生成AIに入力し、検出結果がフィルタ条件のいずれかに該当するかを確認させる。いずれかに該当した検出結果は、報告対象外としてフィルタされる。このように、フィルタ条件生成とフィルタ適用は、生成AIに対する自然言語での指示で実現される。

過検出抑制フィルタは、生成AIが持ち合わせていない報告要否の判断基準をユーザがフィードバックすることで、以降の検出においてその判断基準が適用される仕組みであり、GUI変更検出を継続的に改善することができる。

4. 実装

評価のため、提案方式に基づくプロトタイプを実装した。プロトタイプはPythonで実装し、Large Language Model (LLM)はAzure OpenAI[3]のGPT-4o[4]を使用した。生成AIが関わる機能の実装にはLLMアプリケーション開発向けオープンソースフレームワークLangChain[5]を使用した。管理画面の操作やスクリーンショット取得にはオープンソースモジュールのselenium[6]を使用した。本稿の評価ではAWSマネジメントコンソールから、実際に運用作業で操作することがあるページ39種類を選択し、GUI変更検出を適用した。なお、本稿に掲載したスクリーンショットはいずれもGUI変更検出において2025年1月以前に実際に取得したものである。

5. 評価

5.1 GUI 変更検出例

本稿の評価では、まず、GUI 変更検出技術で検出される AWS マネジメントコンソール上の GUI 変更の実例を示す。GUI の変更が運用作業に影響を与えるかは手順書の内容に依存するが、本評価は運用作業への影響の有無や大きさに関わらず、生成 AI が様々な変更に対する検出結果を示すことを目的としている。

5.1.1 ページ全体の變更

図 4 に、機能の改良に伴いページのレイアウトやデザインが変更されたページの変更前後のスクリーンショットと検出結果を示す。左のスクリーンショットが変更前に取得されたものであり、右のスクリーンショットが変更後に取得されたものである。また、図右側に示したテキストは、GUI 変更検出によって出力された変更内容の説明である。

このページは、AWS Backup で提供されている機能の 1 つであるバックアップポールド[1]が改良され、それに伴い本機能のページが変更されている。この例における変更点は、名称、ページレイアウト、機能に関する説明文の追加など渡るが、GUI 変更検出技術により変更が検出され、過検出抑制フィルタによって除外されることなく、主な変更点が正しく説明されている。



図 4 ページ全体の變更に対する検出結果 [1]

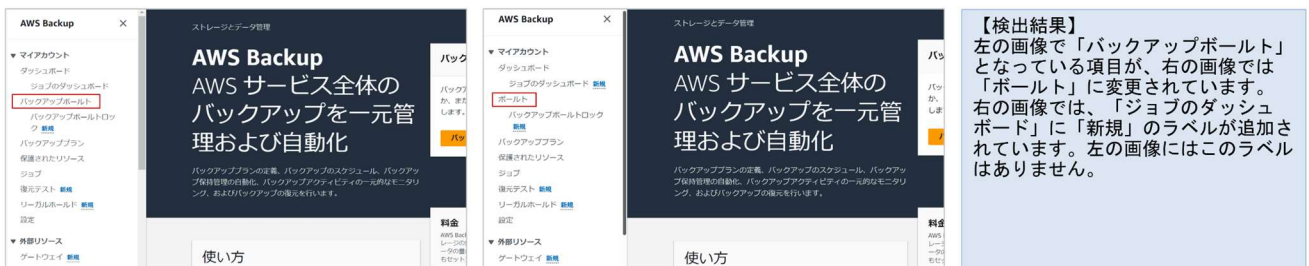


図 5 メニュー上の項目變更に対する検出例 [2]



図 6 表記の變更に対する検出例 [7]

5.1.2 サービスメニュー上の項目變更

図 5 に、サービスメニュー上の機能名が変更されたページの変更前後のスクリーンショットと検出結果を示す。なお赤枠は変更箇所を示している。このページは、5.1.1 節で示した変更に伴い、AWS Backup[2]のページの画面左に表示されるメニュー上の機能表記が「バックアップポールド」から「ポールド」に変更されている。この例に対する検出結果では、機能名が変更されたことが説明され、過検出抑制フィルタによって除外されることなく検出された。

5.1.3 表記の變更

図 6 に、ページ内の表記が変更されたページの変更前後のスクリーンショットと検出結果を示す。このページはスタックという機能の設定画面[7]であり、選択可能なオプションのタイトルが変更され、また各オプションの説明書きが追加されている。この例では、タイトルの変更と説明文の追加が検出され、過検出抑制フィルタによって除外されることなく検出された。

表記の変更は、手順書における説明に大きな影響を与える場合がある。例えばこの例では、手順書に「”テンプレートの準備完了”を選択する。」と記載されているにもかかわらず、操作中のページには”テンプレートの準備完了”というオプションが存在せず、作業時に混乱を招いてしまう。その場合、作業担当者は作業を中断することになるが、GUI 変更検出が有効に作用すれば、手順書の更新や作業担当者への通達など対策を講じることができる。

表 1 生成されたフィルタ条件

#	フィルタ条件
1	通知やお知らせの有無・内容の変化
2	個別のイベント名や件数の増減
3	コンソール上の数字やリソース状態の変化
4	最近アクセスしたサービスや履歴の変化
5	おすすめサービスの項目や順番の変化
6	最終更新日などの具体的な情報の変化

5.2 過検出抑制フィルタ精度評価

本節では、画像差分検出において差分が検出されたサンプルを母集合として、過検出抑制フィルタの判定精度を混合行列に基づく精度を算出する。検出精度評価と同様に、過検出抑制フィルタ機能を有効にした GUI 変更検出 14 回分を対象とする。本評価では、まず過検出抑制フィルタ適用以前に確認された GUI の変更ではない検出結果 19 件に対しフィードバックを与えた。フィードバックの内容としては、通知の有無は報告不要、履歴などは利用に応じて更新されるため報告不要、お知らせやニュースの項目は定期更新されるため報告不要、といった内容である。これらから、3.2 節に前述の処理により 6 つのフィルタ条件が生成された。生成されたフィルタ条件を表 1 に示す。

前述の通り、フィルタ精度評価では GUI 変更検出 14 回分を対象に精度を算出する。まず、フィルタ結果の内訳と精度を表 2 に示す。画像差分検出において 145 件の差分が検出された。その中で、過検出抑制フィルタで定義されている報告不要な検出結果に該当するサンプルは 131 件であった。このうち、過検出抑制フィルタで生成 AI に報告不要と判断された検出結果 (TP) は 112 件であった。また、報告が必要にも関わらず生成 AI によって報告不要となった検出結果 (FP) は 0 件であった。

以上の結果から精度を算出すると、Accuracy は 86.9%、Precision は 100%、Recall は 85.5% である。なお、一般的に FP 及び Precision は検出における過検出に関わる指標で、FP が少なく Precision が大きい場合、過検出が少ないことを意味する。この精度は過検出抑制フィルタの分類精度であり、検出結果においては検出漏れの増減に関わる。そのため、この精度評価の結果は、過検出抑制フィルタにより、見逃しを増やすことなく不要な検出の報告を 85.5%削減できることを示している。

5.3 GUI 変更検出精度評価

GUI 変更検出の精度評価では、過検出抑制フィルタを含めた GUI 変更検出技術の精度を評価するため、15 日分のスクリーンショットを取得の上、過検出抑制フィルタを有効にした GUI 変更検出を 14 回実行し、その結果を対象に、混合行列に基づく精度を算出する。検出対象のページは、4 章に前述の 39 種の Web ページである。提案方式で検出されたサンプルの中で、実際に GUI に変更があったものは True Positive (TP)、変更がなかったものは False Positive (FP) である。FP にはフィルタしきれなかった報告不要な検出結果が含まれる。提案方式で検出されなかったサンプルの中で、実際に GUI 変更がなかったものは True Negative (TN)、実際には変更があったものは False Negative (FN) である。TN には過検出抑制フィルタにより正しく報告不要

表 2 フィルタ精度検証結果

TP	TN	FP	FN	Accuracy	Precision	Recall
112	14	0	19	86.9%	100%	85.5%

表 3 検出精度結果

TP	TN	FP	FN	Accuracy	Precision	Recall
12	513	21	0	96.2%	36.4%	100%

と判断された検出結果が、FN には過検出抑制フィルタにより誤って報告不要と判断された GUI 変更検出結果が含まれる。ただし、フィルタ適用前から差分が検出されていないサンプルを TN、FN に分類する作業は人の目で行うため、算出される精度の値はあくまで参考値とする。検出の内訳と精度を表 3 に示す。14 回の GUI 変更検出の結果、TP は 12 件、TN は 513 件、FP は 21 件、FN は 0 件であった。

以上の結果から精度を算出すると、Accuracy は 96.2%、Precision は 36.4%、Recall は 100% である。全体の検出精度を示す Accuracy は 96.2% と高い精度を示した。FP 及び Precision は検出における過検出に関わる指標で、FP が少なく Precision が大きい場合、過検出が少ないことを意味する。本評価では Precision が低く過検出の割合が高い結果となったが、母数に対し検出対象の数が少ない場合、Precision は低くなりやすい。また、21 件の FP のうち、19 件はフィルタ漏れによるものである。そのため、過検出抑制フィルタの改良により、Precision は改善が期待される。フィルタの精度については 6.1 節で、フィルタで除去できなかった原因の考察については 6.2 節で後述する。また、本評価では GUI 変更検出の FN すなわち検出漏れは 0 件であったが、フィルタを適用する以前に GUI 変更検出を約 30 回実行した中で数件の FN が確認されており、この FN の原因に関する考察を 6.3 節に後述する。

6. 考察

6.1 検出漏れの影響と原因

5.3 節の評価の結果、FN すなわち検出漏れは 0 件であったが、過検出抑制フィルタ適用以前に数件の検出漏れが確認された。GUI 変更検出において検出漏れは、手順書更新の遅れを引き起こし、作業の中断や手順書作成者の負担増大といった影響を及ぼす可能性がある。そこで本節では、検出漏れの実例と原因について考察する。

図 7 に、AWS の Identity and Access Management (IAM) 上にリソースコントロールポリシーという機能が追加された 2 つのページ[8],[9]を示す。両者は同様の変化にもかかわらず、左のページは画像差分検出において変化が検出され、右のページは変化が検出されなかった。右のページでは検出されなかった原因として、画像内の情報量が左のページに比べて多いことが考えられる。左のページはユーザグループの一覧を表示しており、テーブルにはユーザグループが 1 件しか表示されていない。一方右のページはポリシーの一覧を表示しており、画面いっぱいポリシーが表示されている。そのため、画像内に余白の多い左のページでは、生成 AI が画像左側のサービスメニュー上で起きた変化を検出しやすく、一方で余白の少なかった右のページでは同じ変化を検出することができなかったものと思われる。

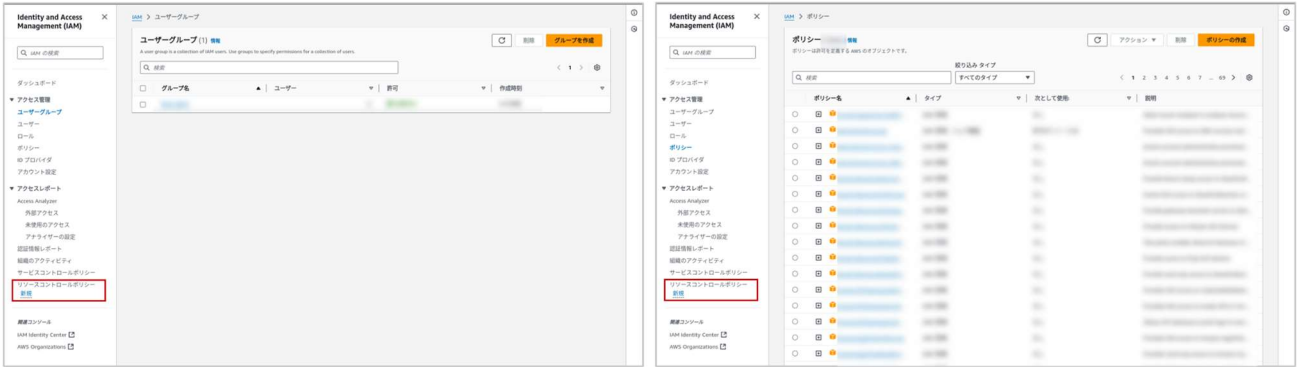


図7 検出されなかった変更の例 [8],[9]



図8 フィルタ漏れした変更の例 [10]

この例のように、AWS のマネジメントコンソール上で、異なるページ間で共通のサービスメニューが表示されているケースは多い。そのため、一部分が共通するページ同士の情報を保持し、一方の検出結果に基づいて、他方でも同様の変更が起きている可能性を通知する仕掛けなどが有効と考えている。ただし、画像差分検出の性能は、使用するモデルの性能に強く依存するため、このような検出漏れは生成 AI の性能向上により解決される可能性も高い。

6.2 フィードバックがフィルタへ与える影響

GUI 変更検出技術の過検出抑制フィルタは、検出結果に対するフィードバックに基づいて適用される。例えば、通知の有無を変更点として検出した場合に、「通知の有無の違いは報告不要」という旨のフィードバックを与えると、その他の検出に対するフィードバックと共に生成 AI がフィルタすべき検出結果の特徴を分析し、「通知の有無や内容の変更は報告不要」といったフィルタ条件が生成される。このように、フィードバックとして与える情報がフィルタ性能に強く影響する。そのため、フィードバックとして与える情報次第では、過剰なフィルタ条件が生成されてしまう恐れがある。例えば、ボタンの形状が四角形から丸角四角形に変化したことを検出した場合に、「ボタンのデザインに関する変更は報告不要」という旨のフィードバックを与え、その旨がフィルタ条件に追加されたとする。このとき、「ボタンのデザイン」という表現は、ボタンの形状だけでなく、色や大きさとも捉えることができ、さらにボタン内の文字も対象と捉えられる。そのため以降の検出時に、ボタンの表記が大きく変化しているのにもかかわらず、過検出抑制フィルタにより報告不要の検出結果と見なされる可能性がある。以上のことから、フィードバックは可能な限り具体的な表現であることが望ましい。また、フィードバックの内容に基づいてフィルタ条件をアップデートする際、問題のあるフィードバックを削除・修正するなどして

メンテナンスする運用が求められるかもしれない。引き続き、この課題の解決に取り組んでいく。

6.3 フィルタ漏れに関する考察

GUI 変更検出技術の過検出抑制フィルタにより検出結果から除外された画像間の差分の中で、最も多かったのが「通知の有無」である。AWS マネジメントコンソールでは、新しい機能のリリース情報や、ユーザに向けて推奨される対応などを青色のオブジェクトなどで表示する。昨日のスクリーンショットには通知が表示されていなかったページにおいて、次の日のスクリーンショットに通知が表示されている場合、画像差分検出はこの差を検出することができる。一方で、通知の有無は GUI の変更ではなく、手順書を更新する必要はない。

生成 AI によるフィルタは、このような通知の有無に関する画像差分を正しくフィルタすることができたが、一部フィルタに失敗したケースがあった。図 8 にフィルタが除去できなかった画像差分検出の例を示す[10]。2 枚の画像の差分は、画面上部の通知の有無である。しかし、通知のオブジェクトによってページ内のボタンが隠れてしまったことで、画像差分検出において生成 AI はボタンがないものと判断し、その旨を差分の説明として出力した。これにより、画像差分の検出結果は通知の有無ではなくボタンの有無となり、過検出抑制フィルタをすり抜けてしまった。このような特殊なケースについても検出件数を削減できるよう、引き続き取り組んでいく。

7. 関連研究

Web サイトの監視技術は、Search Engine Optimization (SEO) 向上を目的としていくつか展開されている。そこで、これらの中から Web サイト監視サービスの Visualping[11]、ChangeTower[12]、Wachete[13]と、オープンソースツールの changd[14]を調査した。これらはいずれも、競合サイト

の監視や、オンラインショッピングの価格監視などに用いられている。

Visualping はカナダの Visualping 社が開発したサービスである。監視したいサイトの URL をアップロードし、表示されるサイトプレビューの中から、監視したい部分をトリミングの要領で指定する。指定範囲に変化があった場合に、事前に指定した連絡先に通知が飛ぶ仕掛けになっている。また、指定範囲において変化した面積の割合を通知条件として指定することもできる。技術の詳細は明らかではないが、同じ条件で画像を取得し機械的に比較していると思われる。そのためあらゆる変化に反応してしまう。近年では AI を活用した機能を利用できるプランも提供されているが、変更監視技術自体はそのままであり、AI は主に UI の充実に寄与している。例えば変更が検出されたページを対象に、変更の内容などを AI に説明させる機能を提供しており、検出結果の確認に役立っている。

ChangeTower も同様に、Web サイトの監視による変更検出サービスを提供している。変更検出にはスクリーンショットと HTML を使用している。Visualping と同様、エリア指定が可能なほか、画像やテキストなど、特定の要素に限定した変更検出も提供している。

Wachete は Wachete 社が提供するサービスで、他のサービスと同様に URL と範囲指定による変更検出を提供している。また、Wachete はパスワード認証が必要なページの監視にも対応している。

オープンソースツールとしては changd が GitHub 上に公開されている。changd は Node.js 上で動作し、Web サイト監視アプリケーションをローカル環境でホストすることができる。

本研究の技術は AWS などのクラウドサービスを対象としており、管理画面の閲覧にはパスワードや Multi-Factor Authentication (MFA) 認証が必要である。Visualping や ChangeTower は認証ページの対応については言及されておらず、パスワード認証に対応している Wachete も MFA といった高度な認証には恐らく対応できない。また、これらのサービスが認証に対応していた場合でも、サービスの提供元に認証に必要な情報を提供することになるため、本研究のユースケースには適さない。Changd はローカル環境でホストできるため認証情報の外部提供のリスクはないが、認証が必要なページに対応していない。

また、これらの既存技術は、画像間の比較に従来の画像比較アルゴリズムを用いている。画像比較アルゴリズムには、ピクセル単位の差を計算する方法や、Structural similarity (SSIM) と呼ばれる画像の類似度を測定する指標を用いる方法がある。またテキストの比較は Web サイトの HTML から文字列を抽出し比較しているものと思われる。一方で本研究の技術では、Web ページ全体を生成 AI に比較させ、生成 AI が GUI の変更と判断した Web ページ内の更を検出し、さらにユーザのフィードバックを反映した生成 AI 活用フィルタで過検出を抑止する。既存技術の中でも通知フィルタ機能を有しているものはあるが、範囲指定や要素指定といった自由度の低い条件であり、生成 AI を活用した自由度の高いフィルタは優位性の 1 つである。

一方で、生成 AI は従来の画像比較アルゴリズムに比べ運用費が高い。本技術では比較的安価な GPT-4o を使用しているが、日々監視する場合コストは嵩む。この問題に対

しては、まず画像比較アルゴリズムによる差分検出か HTML での比較を実行し、それにより差分が検出されたページに対してのみ生成 AI による比較を適用することで、生成 AI の呼び出し回数を削減する解決策を検討している。本件は今後の課題として改良に取り組む予定である。

8. おわりに

本稿では、クラウドサービスの管理画面を定期取得し、画像間差分を生成 AI が検出する機能と、生成 AI を活用し過検出を抑制するフィルタから成る GUI 変更検出技術を提案した。過検出抑制フィルタは、生成 AI を活用し検出へのユーザからのフィードバックを分析し反映することで、FP を削減し検出精度を向上させることができる。提案方式に基づくプロトタイプを実装し、AWS のマネジメントコンソールを対象に検出性能を評価した結果、ビジュアルアップデートや機能名称の変更・追加、説明書きの内容や言語の変更などが検出可能であることを確認した。また精度評価において画像差分検出はマネジメントコンソール上の変化を漏れなく検出し、さらに過検出抑制フィルタにより過検出が 85.5%削減できることを確認した。

商標について

- Amazon Web Services およびその他の AWS 商標

Amazon Web Services およびその他の AWS 商標は、米国およびその他の諸国における Amazon.com, Inc. またはその関連会社の商標である。

- Visualping の商標

Visualping の商標は、米国における WebMonitoring Technologies Inc. の商標である。

- ChangeTower の商標

ChangeTower の商標は、米国、英国および欧州における ChangeTower LLC の商標である。

参考文献

- [1] AWS, "AWS Backup/Vaults," AWS, <https://console.aws.amazon.com/backup/home#/backupvaults> (Last accessed June 6, 2025)
- [2] AWS, "AWS Backup," AWS, [https://console.aws.amazon.com/backup/home#/,](https://console.aws.amazon.com/backup/home#/) (Last accessed June 6, 2025)
- [3] Microsoft, "Azure OpenAI Service," Microsoft, <https://azure.microsoft.com/ja-jp/products/ai-services/openai-service> (Last accessed June 6, 2025)
- [4] OpenAI, "GPT-4o," OpenAI, <https://openai.com/index/hello-gpt-4o/> (Last accessed June 6, 2025)
- [5] LangChain, "LangChain," LangChain, <https://www.langchain.com/> (Last accessed June 6, 2025)
- [6] Selenium, "Selenium," Selenium, <https://www.selenium.dev/> (Last accessed June 6, 2025)
- [7] AWS, "AWS CloudFormation/Stacks/Create stack," AWS, <https://console.aws.amazon.com/cloudformation/home#/stacks/create> (Last accessed June 6, 2025)
- [8] AWS, "AWS/IAM/User groups," AWS, [https://console.aws.amazon.com/iam/home#/groups,](https://console.aws.amazon.com/iam/home#/groups) (Last accessed June 6, 2025)
- [9] AWS, "AWS/IAM/Policies," AWS, <https://console.aws.amazon.com/iam/home#/policies> (Last accessed June 6, 2025)
- [10] AWS, "AWS/IAM/Roles," AWS, <https://console.aws.amazon.com/iam/home#/roles> (Last accessed June 6, 2025)
- [11] Visualping, "Visualping," Visualping, <https://visualping.io/> (Last accessed June 6, 2025)
- [12] ChangeTower, "ChangeTower," ChangeTower, <https://changetower.com/> (Last accessed June 6, 2025)
- [13] Wachete, "Wachete," Wachete, <https://www.wachete.com/> (Last accessed June 6, 2025)
- [14] Paul Aschmann, "changd," Github, <https://github.com/paschmann/changd> (Last accessed June 6, 2025)