

Read It, Don't Watch It: Captioning Bug Recordings Automatically

Sidong Feng[†], Mulong Xie[‡], Yinxing Xue[§], Chunyang Chen^{†*}

[†] Monash University

[‡] Australian National University

[§] University of Science and Technology of China

2023 IEEE/ACM 45th International Conference on
Software Engineering (ICSE), pp.??-??

背景 | バグレポートによるバグ報告

発生したバグの内容や再現手順を記し、バグを報告
—ソフトウェア保守においてバグレポートの処理は重要
—バグの詳細を明確かつ簡潔に記述したものが良いとされる

常に“良いバグレポート”が書かれるとは限らない
原因

- ・ 良いレポートを書くには時間・労力がかかる
 - ・ 専門知識を持たない人による報告 など
- バグ報告のハードルが高い

背景 | 画面録画によるバグ報告

実際に起こったバグを録画し，動画で報告する方法
—iOSやAndroidなどは画面収録の機能が組み込まれている
—より詳細な情報を含められる
→バグ報告のハードルを大幅に下げることができる

問題点：録画からユーザのアクションを理解するのに時間がかかる

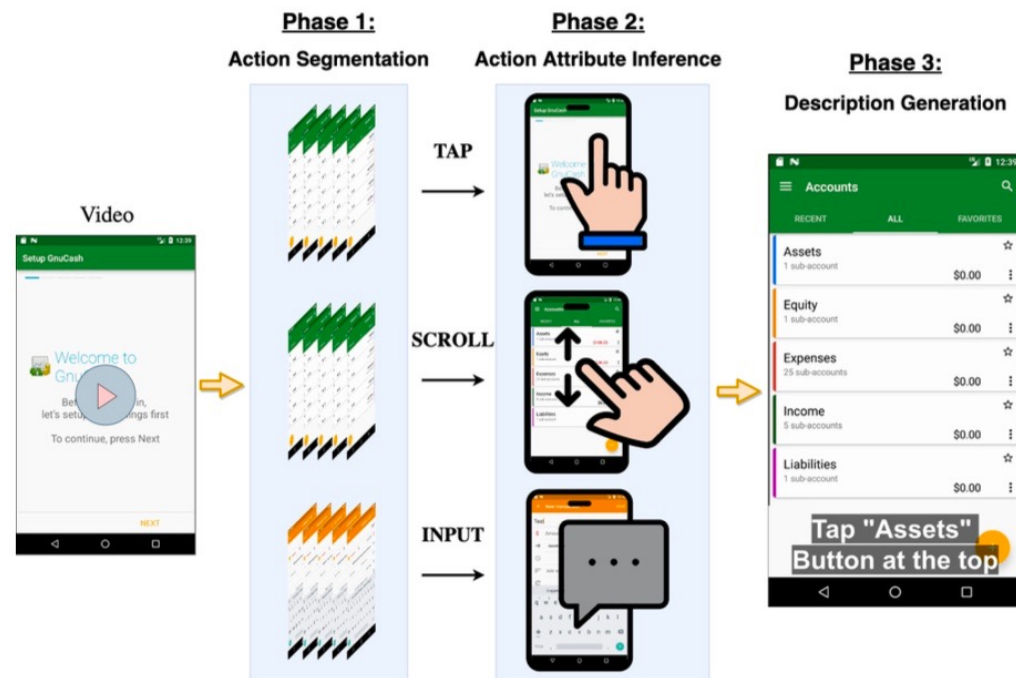
原因

※アクション
タップやスクロール，入力などの動作

- ・ 操作・再生速度が速すぎる
- ・ 映像品質が悪い(画質が悪いなど)
- ・ タップ動作，カーソルなどが目立たない

CAPdroid

- バグ録画におけるユーザのアクションに対して、自動でキャプション(字幕)を生成するアプローチ
- 画像処理技術やDLなど用いて、ユーザアクションの分類やその自然言語記述の生成を行う



目的・アプローチ

研究目的

画面録画によるバグ報告におけるユーザのアクションを，開発者が理解しやすいようにする

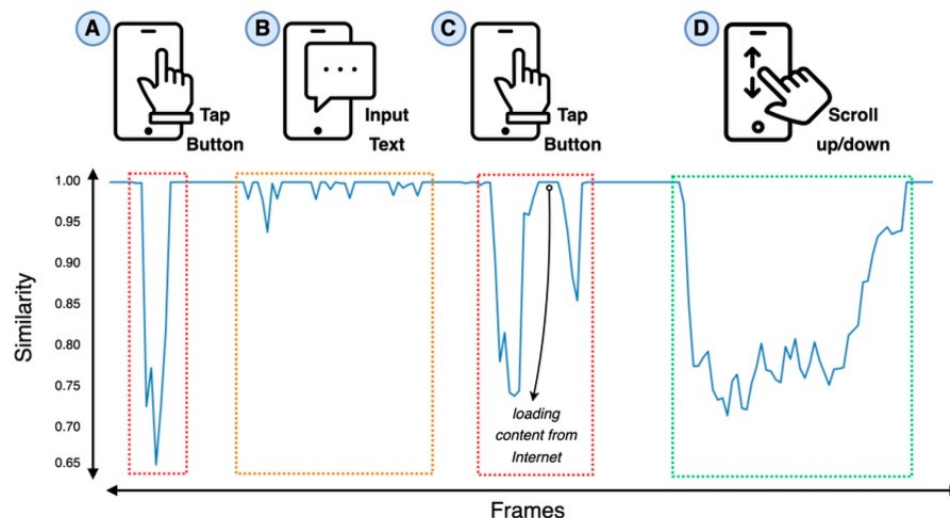
アプローチ

動画内におけるアクションを自動で分析し，その内容を説明する字幕を自動で生成するCAPdroidを開発・評価

CAPdroid概要 | アクションの分類

連続するフレームの差異を数値化し，その類似度にしたがってアクション进行分类

- ①動画をフレームごとに切り分け，それぞれのフレームを輝度に基づき数値化
- ②連続する2フレームを，比較指標SSIMでフレームの類似度を算出(0~1)
- ③類似度の変化の様子を，パターン分析に基づいて3種類のアクション(タップ, スクロール, 入力)に分類



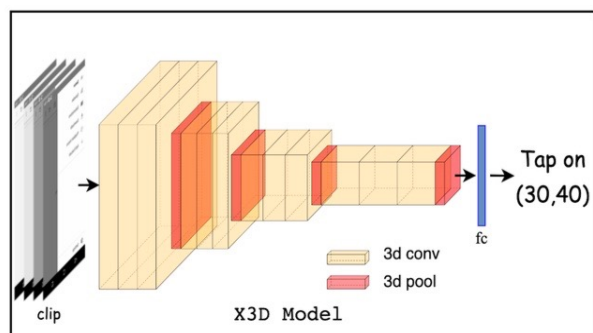
※SSIM：画像の構造的な類似性を算出するメトリクスの一つ

フレーム類似度の変化とアクションの種類

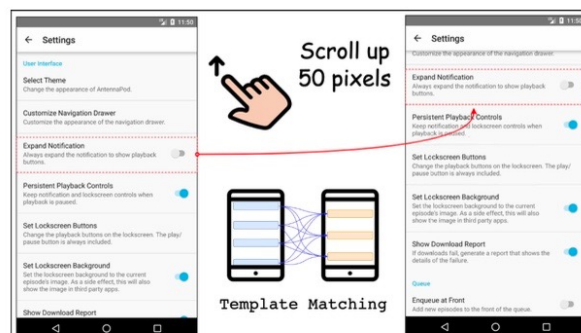
類似度の変化の傾向によって，アクション进行分类できる

CAPdroid概要 | アクション詳細の特定

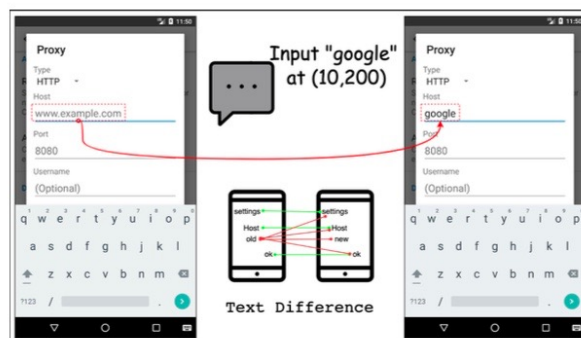
既存の技術を用いて、アクションの詳細(位置や内容)を特定



① **TAP** :
X3D(CNNモデル)を用いて、タップ位置を特定



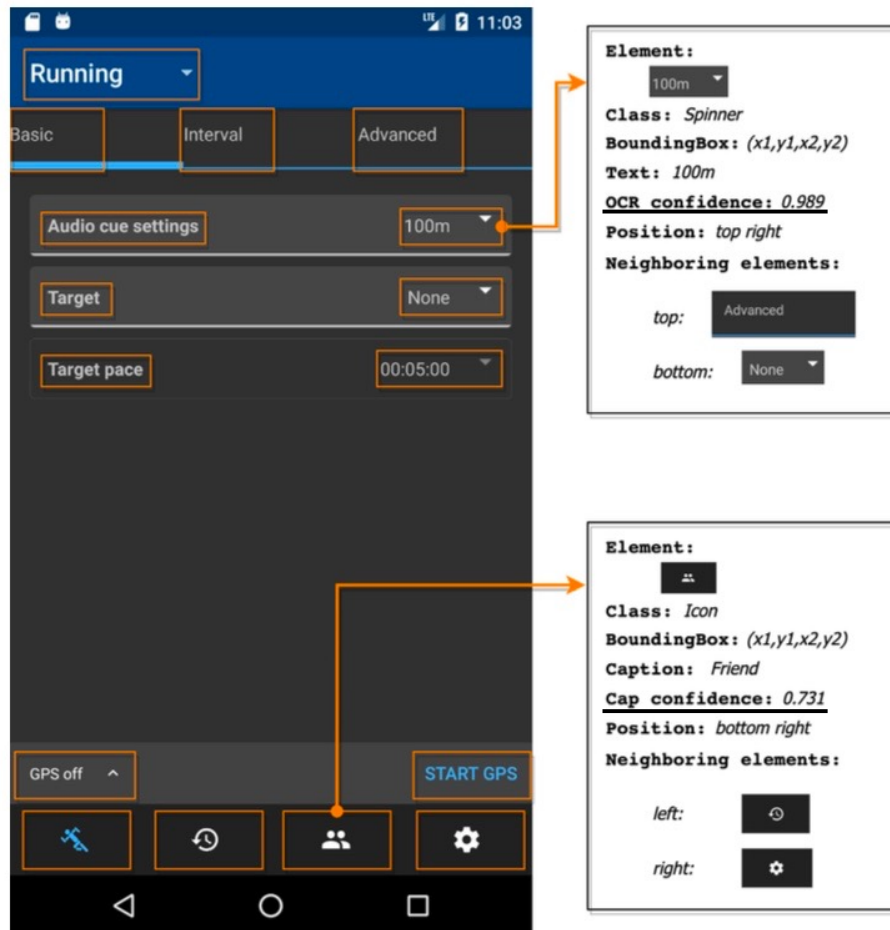
② **SCROLL** :
K-foldテンプレートマッチングでオブジェクトの移動距離を測定し、スクロール距離を特定



③ **INPUT** :
PP-OCRv2(OCR : 光学文字認識)を用いて入力内容を特定

CAPdroid概要 | キャプション生成①

Faster-RCNN(物体検出モデル)で、画面上のオブジェクトの属性や位置などの情報を取得



取得する情報

- Element : 存在するオブジェクト
- Class : オブジェクトの種類
- Text : 書かれている文字
- Caption : アイコンなどの説明
- Confidence : 信頼度
(オブジェクトを的確に検出できていそうか)
- Position : オブジェクトの位置
- Neighboring elements :
近くにあるオブジェクト

CAPdroid概要 | キャプション生成②

物体検出で得られた情報とテンプレート文章をもとに
キャプションを作成

アクション

信頼度などの条件

文書テンプレート

Action	Id	Condition	Template
TAP	1	$(obj_{text/caption} \neq NULL) \wedge (obj_{confid} > \alpha)$	<u>Tap $[obj_{text}]$ $[obj_{class}]$</u>
	2	$(obj_{text/caption} \neq NULL) \wedge (\beta < obj_{confid} < \alpha)$	<u>Tap $[obj_{text}]$ $[obj_{class}]$ at $[obj_{position}]$</u>
	3	$(obj_{text/caption} == NULL) \vee (obj_{confid} < \beta)$	Tap the $[obj_{class}]$ $[nbr_{relation}]$ $[nbr_{text}]$
SCROLL	4	$obj_{text} \neq NULL$	Scroll $[direction]$ $[offset]$ of the screen to $[obj_{text}]$
	5	$obj_{text} == NULL$	Scroll $[direction]$ $[offset]$ of the screen
INPUT	6	$(obj_{text/caption} \neq NULL) \wedge (obj_{confid} > \alpha)$	Input $[text]$ in the $[obj_{text}]$ editText
	7	$(obj_{text} == NULL) \vee (obj_{confid} < \alpha)$	Input $[text]$ in the editText $[nbr_{relation}]$ $[nbr_{text}]$

例：“OK”ボタンをタップ 信頼度=高

→ 「Tap “OK” button」

例：“menu”アイコンをタップ 信頼度=中

→ 「Tap “menu” icon at top left corner」

信頼度が高い・情報が多いものは
単純明快な文章に、
信頼度が低い・情報が少ないものは
情報を追加した文章にするイメージ

CAPdroidの評価

データセット

- ・ Google Playより439のアプリを収集(14カテゴリ, 高評価)
- ・ 自動アプリ探索ツールDroidbotで10分間自動で探索・実行
- ・ その操作を画面録画すると同時に, 正解データとして実際に行ったアクションを抽出
- ・ タッチインジケータによりタップ箇所等を表示

タッチインジケータ: タッチした場所に円などのアニメーションを表示



デフォルト(白円): 181件



カーソル(矢印): 152件



カスタム: 106件

RQ1 | アクション分類の正確性

収集した録画のアクションを提案手法により分類, その精度をF1値, Accuracyで評価

Method	TAP		SCROLL		INPUT		Overall	
	VS	Acc	VS	Acc	VS	Acc	VS	Acc
ABS	0.56	0.69	0.59	0.69	0.67	0.73	0.61	0.71
HIST	0.71	0.80	0.62	0.71	0.75	0.84	0.70	0.79
SIFT	0.61	0.71	0.60	0.73	0.63	0.79	0.62	0.75
SURF	0.55	0.71	0.59	0.72	0.60	0.77	0.58	0.74
EDGE	0.61	0.75	0.55	0.70	0.66	0.78	0.61	0.75
Ours	0.81	0.89	0.83	0.92	0.90	0.97	0.84	0.93

VS : video segmentation F1-score
Acc : Accuracy

Ours : SSIM
その他 : その他の類似性メトリクス

SSIMを用いた提案手法はベースライン手法と比べ優れている
→SSIMは空間とピクセルの多くの側面から類似性の測定を行うため

RQ2 | アクション詳細の正確性

TAP位置やSCROLL距離, INPUTされた文字の Accuracyを測定

Methods	TAP			SCROLL			INPUT			Overall
	default	cursor	custom	default	cursor	custom	default	cursor	custom	
V2S [8]	84.19%	69.66%	36.10%	85.19%	63.31%	29.00%	-	-	-	61.24%
GIFdroid [4]	85.78%	88.01%	87.16%	72.84%	71.01%	69.77%	35.13%	32.11%	28.39%	63.35%
CAPdroid	91.06%	90.28%	92.67%	94.87%	94.63%	95.12%	87.86%	88.62%	88.11%	91.46%

従来手法(上2つ)に比べてCAPdroidははるかに正確に推定できている

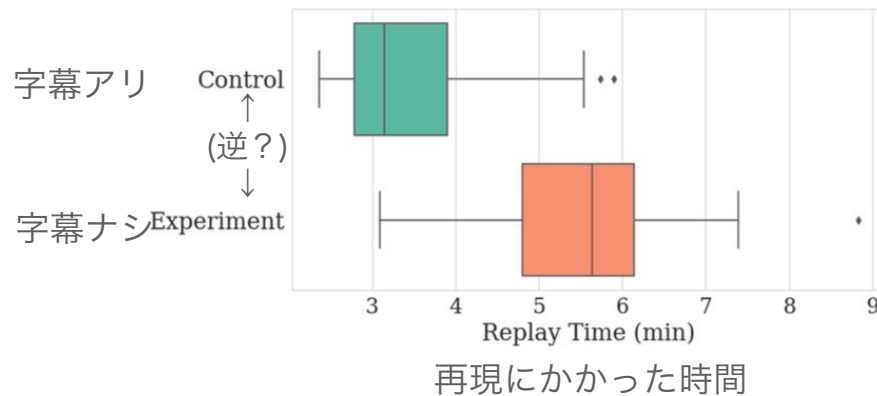
→空間的・時間的特徴を用いている, インジケータに依存しない, 入力テキストの推定が正確なため

CAPdroidの有用性に関する調査①

CAPdroidによって生成されたキャプションが実際に役に立つかどうかを調査

- ・参加者：8人(学生と開発者)
- ・バグ報告の動画を見て，バグを再現する時間を測定
- ・4人は字幕なし，残り4人は字幕あり

結果



字幕アリの方が，バグの再現にかかった時間が短かった
(平均3.46分 vs 5.33分)
→生成された字幕によって，アクションがわかりやすくなっている

CAPdroidの有用性に関する調査②

タスクの後、バグ録画に関するフィードバックを収集
説明の明瞭さ、簡潔さ、有用性について5段階評価でアンケート

結果

	字幕ナシ	字幕アリ
Measures	Control	Experiment
Clearness	2.50	4.25*
Conciseness	1.75	4.50*
Usefulness	-	4.75

字幕アリの方が全体的に大幅高評価

→CAPdroidの有用性を示唆

まとめ

| 画面録画によるバグ報告におけるアクションの理解を、
キャプションの自動生成により促進するCAPdroidを
提案

| 画像処理技術等を用いてアクションの分類や推定、得
られた情報による字幕の生成を行う

| 実験によりバグ再現における有用性が示された

所感

| 画像認識技術に関する用語が難しい

| 予期せぬ画面の変化があった際のアクション分類？

→フレームの類似度で分類するなら，広告や動画が映った際は？

| 画像認識技術のソフトウェア工学への応用は新鮮に感じた

→他のタスクでも使えそう