# tRaNSLAtinG VIDEO RECORDING OF MOBILE APP USAGE IN TO REPLAYABLE SCENARIOS

The author of this title is **“Carlos Bernal-Cardenas”, “Nathan Cooper”, “Kevin Moran”, “Oscar Chaparro”, “Andrian Marcus”, “Denys Poshyvanyk”.** It`s conference name is **“International Conference on Software Engineering”** (ICSE’20), it was published on “**May 23-29, 2020**”at “**Seoul**, republic of **Korea”**.

**INTRODUCTION:**

These information sources include user reviews, crash reports, bug reports, and emails, among others. An increasingly common component of these software artifacts is graphical information, such as screenshots or screen recordings. In fact, many crowd-testing and bug reporting frameworks have built-in screen recording features to help developers collect mobile application usage data and faults [6, 7, 21, 22].

1. Help understand how users interact with apps [14, 68].
2. Process bug reports and feature requests from end-users [27].
3. Aid in bug comprehension for testing related tasks [53].

The manual effort required by this comprehension process complicates a development workflow that is already constrained by language dichotomies [55] and several challenges unique to mobile software, including:

1. Pressure for frequent releases [39, 43].
2. Rapidly evolving platforms and APIs [24, 49].
3. Constant noisy feedback from users [30, 31, 62–64].
4. Fragmentation in the mobile device ecosystem [1, 37, 72] among others [51].

To improve and automate the analysis of video-related mobile development artifacts, we introduce Video to Scenario (V2S), a lightweight automated approach for translating video screen recordings of Android app usages into replayable scenarios. V2S adapts recent Deep Learning (DL) models for object detection and image classification to accurately detect and classify different types of user actions performed on the screen. We conducted a comprehensive evaluation of V2S using both videos collected from users reproducing bugs as well as general usage videos from the top-rated apps of 32 categories in the Google Play market.

The results of our evaluation indicate that V2S is accurate, and is able to correctly reproduce 89% of events across collected videos. In summary, the main contributions of our work are as follows:

1. V2S, the first record-and-replay approach for Android that functions purely on screen-recordings of app usages.
2. An automated pipeline for dataset generation and model training to identify user interactions from screen recordings.
3. The results of an extensive empirical evaluation of V2S that measures the accuracy, robustness, and efficiency across 175 videos from 80 applications.
4. The results of a case study with three industrial partners, who develop commercial apps, highlighting V2S’s potential usefulness, as well as areas for improvement and extension.
5. An online appendix [26], which contains examples of videos replayed by V2S, experimental data, source code, trained models, and our evaluation.

**THE V2S APPROACH:**

V2S’s architecture, which is divided into three main phases:

**Touch Detection phase**, which identifies user touches in each frame of an input video.

**Action Classification phase** that groups and classifies the detected touches into discrete user actions (i.e., tap, long-tap, and swipe).

**Scenario Generation phase** that exports and formats these actions into a replayable script.

**METHODOLOGY:**

we describe the procedure we used to evaluate V2S.

The main quality focus of our study is the extent to which V2S can generate replayable scenarios that mimic original user GUI inputs. To achieve our study goals, we formulated the following five research questions:

• RQ1: How accurate is V2S in identifying the location of the touch indicator?

• RQ2: How accurate is V2S in identifying the opacity of the touch indicator?

• RQ3: How effective is V2S in generating a sequence of events that accurately mimics the user behavior from video recordings of different applications?

• RQ4: What is V2S’s overhead in terms of scenario generation?

• RQ5: Do practitioners perceive V2S as useful?

**Accuracy of Faster R-CNN:**

We first evaluated the ability of V2S’s Faster RCNN to accurately identify and localize the touch indicators present in screen recording frames with bounding boxes.

**Accuracy of Opacity CNN:**

We evaluated the ability of V2S’s Opacity CNN to predict whether the opacity of the touch indicator is solid or semitransparent.

**Accuracy on Different Scenarios:**

we carried out two studies designed to assess both the depth, and breadth of V2S’s abilities to reproduce user events depicted in screen recordings. The first, Controlled Study, measures the depth of V2S’s abilities through a user study during which we collected real videos from end users depicting: bugs, real crashes, synthetically injected crashes, and normal usage scenarios for 20 apps. Next in the Popular Applications Study we measured the breadth of V2S’s abilities by recording scenarios for a larger, more diverse set of 64 most popular apps from the Google Play.

**Performance:**

To investigate RQ4, we evaluated V2S by calculating the average time it takes for a video to pass through each of the three phases of the V2S approach on commodity hardware (i.e., a single NVIDIA GTX 1080Ti). We see this as a worst-case scenario for V2S performance, as our approach could perform substantially faster on specialized hardware.

**Perceived Usefulness:**

Ultimately, our goal is to integrate V2S into real-world development environments. Thus, as part of our evaluation, we investigated V2S’s perceived usefulness with three developers who build Android apps (or web apps for mobile) for their respective companies.

**CONCLUSION:**

We have presented V2S, an approach for automatically translating video recordings of Android app usages into replayable scenarios. A comprehensive evaluation indicates that V2S: Accurately identifies touch indicators and it is able to differentiate between opacity levels, is capable of reproducing a high percentage of complete scenarios related to crashes and other bugs, with promising results for general user scenarios as well, and is potentially useful to support real developers during a variety of tasks.