Technical Disclosure Commons

Defensive Publications Series

November 2019

PREDICTED APPLICATION PRELOAD BASED ON CONTEXT

Collin Irwin

Rachel Hausmann

Joe Rothlin

PREDICTED APPLICATION PRELOAD BASED ON CONTEXT

ABSTRACT

A computing device (e.g., a mobile phone, camera, tablet computer, etc.) uses an application preload prediction model (e.g., an artificial intelligence model) for preloading an application. The computing device may execute an application that employs a user interface to facilitate human-machine interaction. The computing device may collect contextual user data and use a prediction model (e.g., an application preload prediction module) to analyze the collected contextual data to determine and predict an application to preload in background mode even with the computing device's screen locked. For example, a user device (e.g. mobile phone) with a touch screen predicts a user's intention to use the device and intelligently preloads applications (apps) (e.g., camera app, browser, terminal application, map, virtual assistant, emergency dialer, etc.) in the background of the device, even when the screen is off. The device includes a machine learned model that predicts the user's intention to use the device based on the information from one or more sensors (such as light sensor, accelerometer, gyroscope, GPS, and proximity sensor) and location information, and preloads the necessary app before the user unlocks the screen or opens the app. This allows for rapid launching of the preloaded app in response to the user selecting the app.

DESCRIPTION

A user may interact with applications executing on a computing device (e.g., a mobile phone, tablet computer, smart phone, desktop computer, or the like). In some examples, a computing device may include or communicate with a touch-sensitive display that may enable a user to unlock the computing device from a locked mode, and interact with applications executing on the computing device. Some applications may provide functionality through receiving indications of various user inputs, such as key taps.

Applications can sometimes take too long to load when a mobile device, such as a smart phone, is unlocked and used after being in a state in which the screen is off or the device is in a power save mode. For example, users may be frustrated by a camera application on a mobile device taking seconds to load when it is urgently required. This document describes techniques for preloading applications that may allow for quick launching of a desired application, as well as improving general lock screen performance. The techniques may also be applied to applications other than camera applications, such as a map application, a virtual assistant application, an emergency dialer application, for example. Through use of phone sensors (light level, accelerometer, gyroscope, proximity) and other information (location), an application preload prediction model may use machine learning techniques to smartly predict a user intention for their device before the screen or any buttons are directly interacted with.

Users frequently draw their phones from their pockets and quickly open the camera app to take photos. Often, time is of the essence as the intended subject is in motion and may quickly disappear from view (e.g. a cute moment at a family gathering or a wildlife sighting). Missing a photo opportunity due to slow camera loading can be frustrating. Phones with slow camera load times are often disparaged by the public and the media.

In some examples, the techniques described herein may allow a mobile device such as a mobile phone to detect a user's intention to take a photo by detecting one or more potential cues that indicate a likelihood of imminent use of the camera application. The camera app can then be intelligently preloaded in the background, potentially even while the screen is off, for significantly quicker rendering and shorter time to interactivity for the user.

Figure 1 illustrates a mobile device having a user interface (UI) device, an output device, and an application preload prediction module. Example mobile devices include mobile phones, tablets, digital cameras, laptops, gaming systems, e-book readers, televisions, wearable computing devices, or any other type of mobile computing device. The mobile device may have installed multiple applications ("apps") (e.g., camera app, browser, terminal application, Google assistant, map, virtual assistant, etc.).

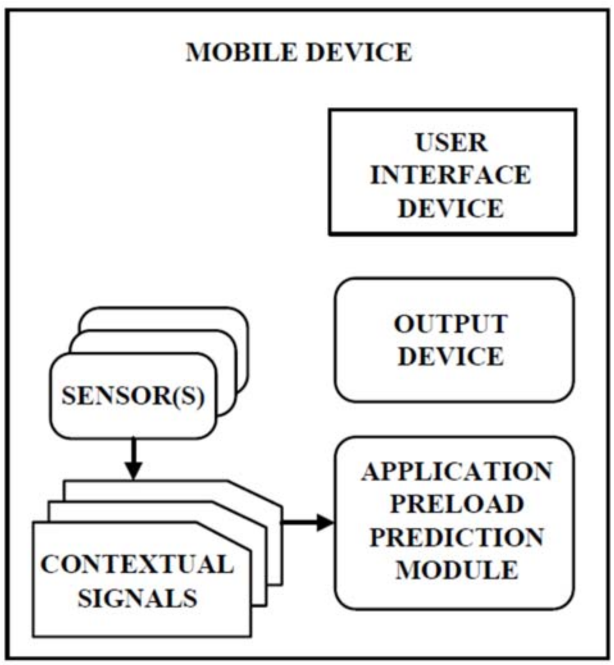

FIG. 1

The user interface device may function as an input device for the mobile device, such as using a presence-sensitive input screen. The output device may function as an output (e.g., display) device using one or more display devices, such as a liquid crystal display (LCD), dot matrix display, light emitting diode (LED) display, organic light-emitting diode (OLED) display, e-ink, or similar monochrome or color display capable of outputting visible information to a user.

The user interface device may control a graphical user interface including determining what the user interface device presents and what information is exchanged between the user interface device and other applications or components (e.g., the output device and the application preload prediction module) of the mobile device. For example, the user interface device may receive information from a component of the mobile device for generating a user interface and elements thereof. In response, the user interface device may output instructions and information, causing the graphical user interface to display an application. The application preload prediction module may receive information from the user interface device in response to inputs detected at locations of a screen of the user interface device at which elements of the graphical user interface are displayed. The user interface device disseminates information about inputs detected by the graphical user interface to other components of the mobile device for interpreting the inputs.

The user interface device may cause the graphical user interface to facilitate interaction between users and the mobile device. For example, when in a locked mode, the user interface device may cause the graphical user interface not to display anything (i.e., the screen is off), or the user interface device may cause the graphical user interface to display a lockscreen but may provide limited access to functionality of the mobile device. When in an unlocked mode, the mobile device may provide access to greater functionality of the mobile device than when in the locked mode. For example, the user interface device may cause the graphical user interface to display a menu of applications installed on the mobile device, including icons that a user may select to launch an application (e.g., a camera application).

The application preload prediction module uses a prediction engine that applies a context model. The context model may be a machine-learned model applied by the prediction engine of application preload prediction module. The machine-learned model may be a machine learning model trained using classifiers that correlate one or more sensed signals, such as device sensor signals indicative of user actions, device movement, and device orientation, with corresponding mobile applications. The machine learned model may be trained using multiple different classifiers for multiple different applications. The application preload prediction module uses the machine-learned model to predetermine one or more applications a user of the mobile device intends to use, allowing for pre-launching of the mobile device when the mobile device is in a locked mode.

The example mobile device includes one or more sensors that detect sensed conditions associated with the mobile device. The one or more sensors may include a gyroscope, proximity sensor, accelerometer, light sensor, GPS, or other type of sensor. The mobile device may collect data indicative of contextual signals, including data from the sensors, which may in some examples be preprocessed, correlated, and interpreted.

As examples, the contextual signals that may form the input data (state) for an application preload prediction algorithm used by application preload prediction module may include signals indicative of: removal of mobile device from pocket; sudden acceleration or change in device orientation; moving a mobile device into horizontal (landscape) orientation; orienting a mobile device in unusual direction; bringing mobile device into alignment with eyes or head; change in light level; decrease in proximity to a surface, application usage data, location history, or other cues. For example, users may be likely to intend to take a photograph when the mobile device is quickly moved to a horizontal (landscape) orientation. As another example, users are likely to intend to take a photograph when aiming in abnormal directions such as an overhead angle and displacement. Moreover, users are likely to align the mobile device with their eyes when intending to take a photograph. Users often bring the mobile device to near-head height, significantly higher than normal use. As a further example, detected cues may include a sudden light level increase surrounding the mobile device, and/or proximity decrease, such as when the mobile device transitions from being in a pocket in which the surface of the pocket was close to the mobile device and then the mobile device is in open air and is no longer proximal to a surface. In some examples, application usage history and/or location data may indicate locations in which particular applications are frequently used (e.g., camera applications), either by a given user based on their own usage history or based on aggregate user data.

As described herein, the mobile device collects the contextual signals and the application preload prediction module analyzes the contextual signals only after receiving explicit authorization from the user to do so. After receiving such authorization, the mobile device and the application preload prediction module may begin to collect and analyze the contextual signals.

In some examples, the mobile device may employ machine learning technology in the application preload prediction module and may include neural networks (not shown) that may ingest the contextual signals. In some examples, the application preload prediction module may include a neural net module that may ingest the contextual signals. In other examples, the mobile device may use a hardcoded analysis of the contextual signals.

Based on analyzing the contextual signals, the application preload prediction module identifies a predicted application (e.g., above a certain confidence level threshold), and causes the mobile device to preload the predicted application. For example, it may not be worth the performance cost to do a full preloading of the camera app every time a phone is moved. A full loading and pre-rendering of the app and the camera's output can then be done and provided to the user interface device. In this manner, the techniques described herein operate in an intelligent way to determine which apps to preload. The mobile device is enabled to load needed apps and may avoid loading unwanted apps.

The techniques described herein can be applied to accident (vehicular crash) detection. Another application could be for fall detection (both phone falls and user falls, such as tripping), which can be useful for certain user populations, such as elderly users and those with physical ability differences. As an example, the device also detects accident (i.e., vehicular crash) and fall of both the user and the device (i.e., tripping) based on the sensor and location information, and preloads the necessary app (e.g., emergency dialer). For example, the application preload prediction module may determine, based on accelerometer data and/or other data, that an emergency dialer application should be preloaded even before the user unlocks the mobile device. As a further example, the application preload prediction module may automatically open a map application upon detecting cues that may indicate a user may be likely to use a map application, such as location data (indicating user is away from a home location), accelerometer or other data (indicating user is driving or has entered a car).

In some examples, the mobile device implements a fast execution computing framework to support the application preload prediction module, running effectively as an additional layer. This can be done by integrating the application preload prediction system at the operating system level of the mobile device, having a lightweight model architecture (e.g., only a few layers) that is more easily deployed on devices with some hardware acceleration for neural nets.

The techniques described herein could be applied at the application level in some examples, instead of at the operating system level.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable collection of user information (e.g., information about a user's social network, social actions, or screenshots of the user's screen), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

References:

WO 2018144014 A1

U.S. Patent No. 9,189,252 B2