December 2019

Improving Squeeze Gesture Detection Through Machine Learning

N/A

Improving Squeeze Gesture Detection Through Machine Learning

Abstract:

This publication describes techniques and methods, implemented on a mobile device, to improve squeeze gesture detection. The technique incorporates the utilization of a machine-learned model that can more precisely determine whether a user squeezed the mobile device with the intent of interaction (*e.g.*, invoke an intelligent virtual assistant user interface, launch applications, dismiss notifications).

Keywords:

Smartphone, squeeze gesture detection, heuristic algorithm, machine learning, machine-learned model, intelligent virtual assistant, user interface, gesture recognition

Background:

Mobile devices are advancing in their ability to perceive user input. In one such instance, a mobile device, such as a smartphone, can detect squeeze gestures (*e.g.*, a type of gesture in which a user can grasp and squeeze the sides of the mobile device as a method of interaction with the mobile device). For instance, a user (Jane) may desire to activate an intelligent virtual assistant by squeezing the sides of her mobile device.

Current implementations of squeeze gesture detection may rely on strain gauge sensors that detect a squeeze, measure the force of the squeeze, and generate a corresponding electrical signal (squeeze signal). The squeeze signal may then be passed through a heuristic pipeline containing data reduction and refining features along with a heuristic algorithm. The heuristic algorithm is a

simple, yet effective, function that can quickly analyze incoming signals and determine either a detected squeeze signal represents a true squeeze (*e.g.*, an intentional squeeze from the user) or a false squeeze (*e.g.*, a squeeze caused by a protective casing of the mobile device, an inadvertent squeeze). If the heuristic algorithm determines that the squeeze signal represents a true squeeze, then the algorithm can direct the operating system (OS) to provide feedback to the user, for instance providing a haptic response (*e.g.*, device vibration feedback to the user indicating that the OS detected a gesture) and perform an action (*e.g.*, triggering an intelligent virtual assistant, launching an application, dismissing a notification). Conversely, if the heuristic algorithm determines that the squeeze signal represents a false squeeze, then no haptic feedback is produced, nor are any actions performed.

Figure 1, below, illustrates a flowchart demonstrating a current computing architecture for squeeze gesture detection.

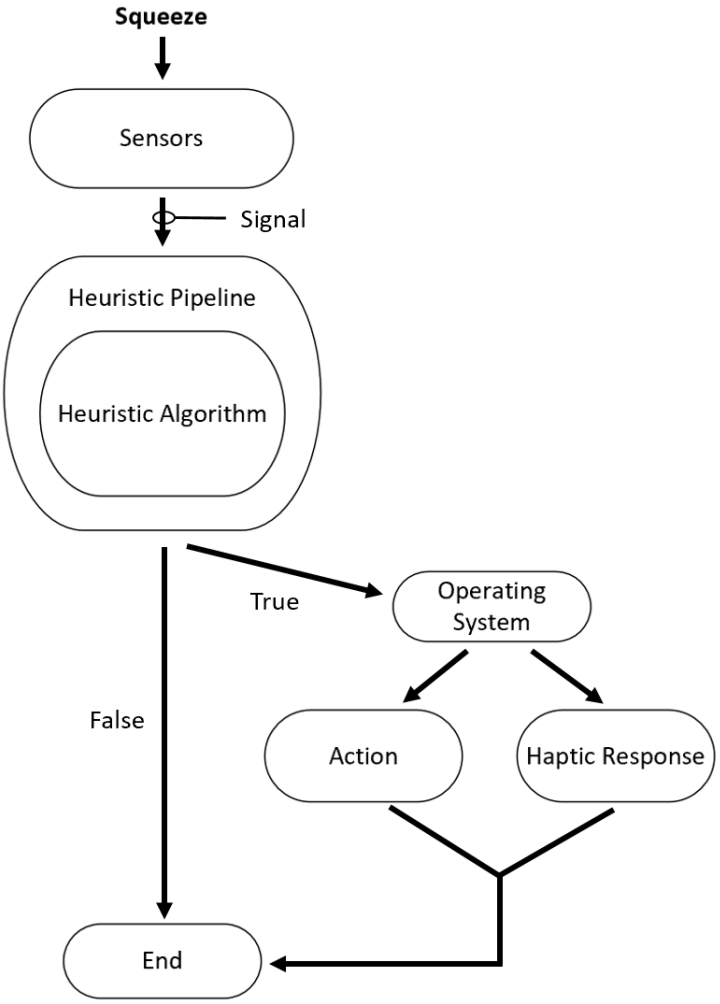

Figure 1

As illustrated in Figure 1, the squeeze sensors on the mobile device detected a squeeze and generated a squeeze signal. The squeeze signal is passed through the heuristic pipeline and analyzed by the heuristic algorithm to determine whether the detected squeeze represents a true or false squeeze. If the heuristic algorithm determines that the squeeze signal indicates a false squeeze, then a haptic response is not produced, nor are any actions performed. If, on the other hand, the heuristic algorithm determines that the squeeze signal indicates a true squeeze, then the OS provides haptic feedback to the user and performs an action.

In utilizing heuristics, some indicators (*e.g.*, patterns, sequences) in the squeeze signal may be overlooked that could otherwise assist in more precisely determining the existence of a true or

false squeeze. Additionally, not every squeeze input from the user is meant to be interpreted as an attempt to interact with the mobile device. For instance, a user may squeeze the mobile device when plugging in a charger, without intending to input a squeeze gesture. Therefore, it is desirable to incorporate machine learning into the heuristic pipeline to extract indicators from a squeeze signal that conventional heuristics may ignore, to the end that squeeze intent may be determined.

Description:

This publication describes techniques and methods, implemented on a mobile device, to improve squeeze gesture detection. The techniques incorporate the utilization of a machine-learned model (ML model) that can determine not only true squeezes (*e.g.*, a squeeze from a user) from false squeezes (*e.g.*, squeezing from a protective casing), but also squeeze intent, such as whether a user squeezed the mobile device with the intent of interaction (*e.g.*, notification dismissal, application launching, intelligent virtual assistant user interface (UI) invocation). The methods include steps directed at enhancing the usability of squeeze gesture detection on a mobile device. An exemplary mobile device, such as a smartphone, includes a display, sensors (*e.g.*, strain gauge sensors, a microphone), a haptic mechanism, a processor, and a computer-readable medium (CRM). The CRM may include the operating system (OS) of the mobile device and an ML model.

The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable-length sequences. The ML model may be iteratively trained, off-device, to analyze strain gauge sensor signals (squeeze signals) and determine squeeze intent. After sufficient training, the ML model can be deployed to the CRM.

Once installed on the mobile device, the ML model can be embedded in the heuristic pipeline to function as the core decision engine in determining true and false squeezes, along with squeeze intent. Figure 2 illustrates a squeeze gesture detection architecture in a flowchart.

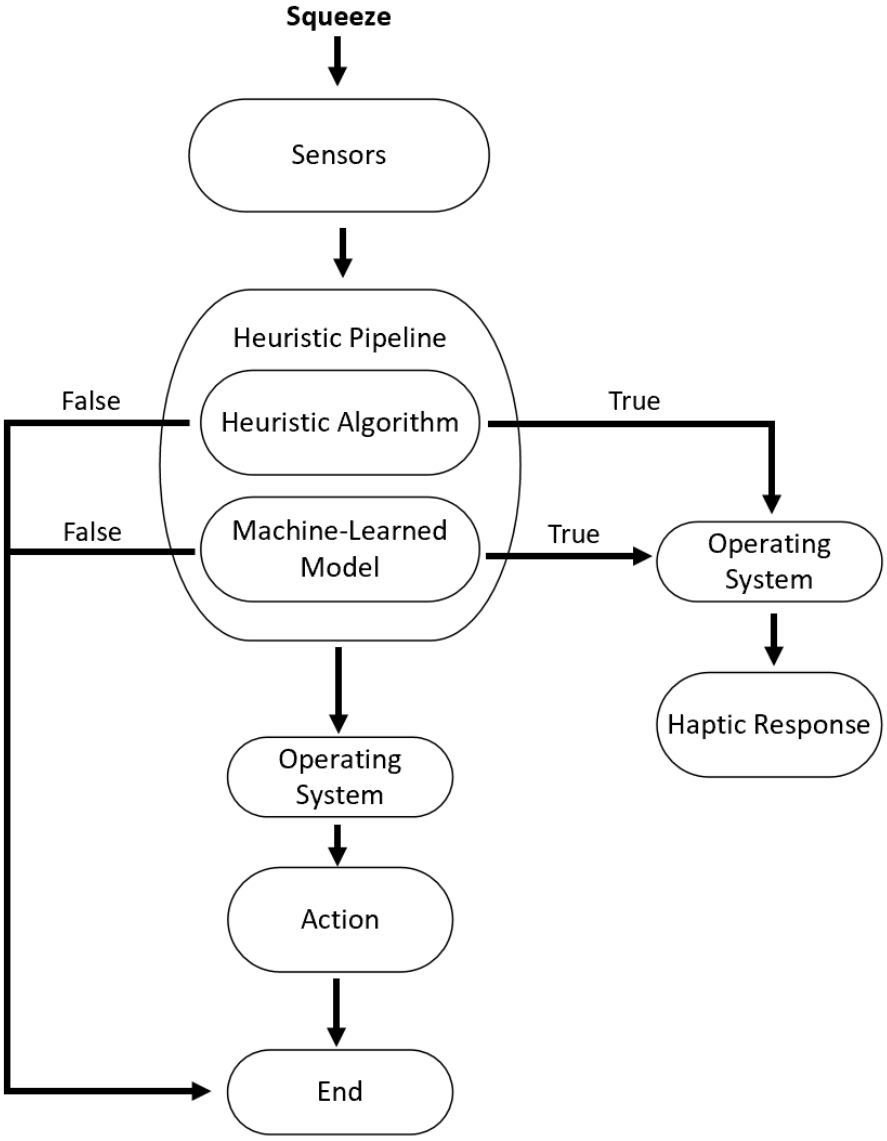

Figure 2

As illustrated in Figure 2, a squeeze is measured by a device sensor, and a squeeze signal is generated. The squeeze signal is then passed through the heuristic pipeline, where it is first analyzed by the heuristic algorithm. If the heuristic algorithm determines that the detected squeeze

represents a false squeeze, then no further actions are triggered. If, however, the heuristic algorithm determines that the detected squeeze represents a true squeeze, then the OS produces a first haptic response. The squeeze signal is then analyzed by the ML model. If the ML model determines that the detected squeeze represents a false squeeze, then no further actions are triggered. If, on the other hand, the ML model determines that the detected squeeze represents a true squeeze, then the OS produces a second haptic response. If both the heuristic algorithm and ML model determine that the detected squeeze represents a true squeeze, then an action is performed, such as the triggering of an intelligent virtual assistant, application launching, or notification dismissal.

The implementation of the novel squeeze gesture detection architecture on a mobile device can be realized in the following example. A user (Jane) may desire to trigger or otherwise activate the user interface of an intelligent virtual assistant on her mobile device. To do so, she simply squeezes the sides of her mobile device. The pressure exerted on the mobile device by Jane's hand is measured by the squeeze sensors, a squeeze signal is generated, and the squeeze signal is analyzed by the heuristic algorithm. After a squeeze has been identified in the squeeze signal, the first haptic response is produced. Next, the ML model analyzes the squeeze signal, determines a true squeeze, and the second haptic response is produced. Lastly, the OS triggers the user interface of the intelligent virtual assistant.

With the ML model integrated with the heuristic pipeline subsequent to the heuristic algorithm, the ML model functions as the core decision engine in differentiating true squeezes and false squeezes, along with squeeze intent. As a result of this configuration, two squeeze gesture detection benefits are evident. First, true squeezes and false squeezes are more precisely identified. Second, the heuristic algorithm and the ML model, along with their respective haptic responses,

introduce a concept of "partial" squeeze gestures. A partial squeeze gesture occurs when both (1) a user input is detected, the heuristic algorithm determines a true squeeze exists, and the heuristic algorithm triggers a haptic response, and (2) the ML model determines a false squeeze and does not produce a second haptic response.

The advantage of partial squeeze gestures can be appreciated in the following example. While plugging a charger into her phone, Jane firmly grips the sides of her mobile device. The heuristic algorithm interprets the resulting squeeze signal as a true squeeze and produces a haptic response, while the ML model, on the other hand, determines that the squeeze signal represents a false squeeze and does not produce a haptic response. In events like these, triggering squeeze gesture detection is undesirable; otherwise, an undesired action may be performed by the mobile device. Consequently, a partial squeeze gesture haptic response may reassure the user that no actions have been triggered.

In addition to the descriptions above, squeeze gesture sensitivity settings for the mobile device may be mapped to the ML model's squeeze gesture detection thresholds with non-linear transformations so that the user may configure squeeze gesture sensitivity to their preference. For example, one user may desire to trigger squeeze gesture detection by means of a gentle squeeze, while another user may desire to trigger squeeze gesture detection with a firmer squeeze.

Finally, when a squeeze is detected and confirmed by the user through immediate interaction with the mobile device following activation, the captured sensor data may be assigned a positive label and fed back to the ML model as further training for validation of future squeezes. For instance, Jane activates an intelligent virtual assistant by squeezing her mobile device and immediately providing a voice command detected by the microphone. Both the confirmation and immediacy of the subsequent interaction can reaffirm the ML model's decision making. As a

result, the measured squeeze signal can be assigned a positive label and fed back to the training pipeline to re-train the model. In this way, squeeze gesture detection is enhanced through the personalization of the ML model.

In conclusion, the ML model cooperatively operating inside the heuristic pipeline as the core decision engine for determining squeeze intent can enhance squeeze gesture detection by more precisely determining true and false squeezes along with squeeze intent. Furthermore, the usability of squeeze gesture detection is improved by the introduction of partial squeezes gestures and through personalization of the ML model.

References:

[1] Negulescu, Matei & Ruiz, Jaime & Lank, Edward. (2012). A Recognition Safety Net: Bi-Level Threshold Recognition for Mobile Motion Gestures. 10.1145/2371574.2371598. https://www.researchgate.net/publication/266064250_A_Recognition_Safety_Net_Bi-Level_Threshold_Recognition_for_Mobile_Motion_Gestures.