Wearable machine learning for recognizing and controlling smart devices

Conference Paper

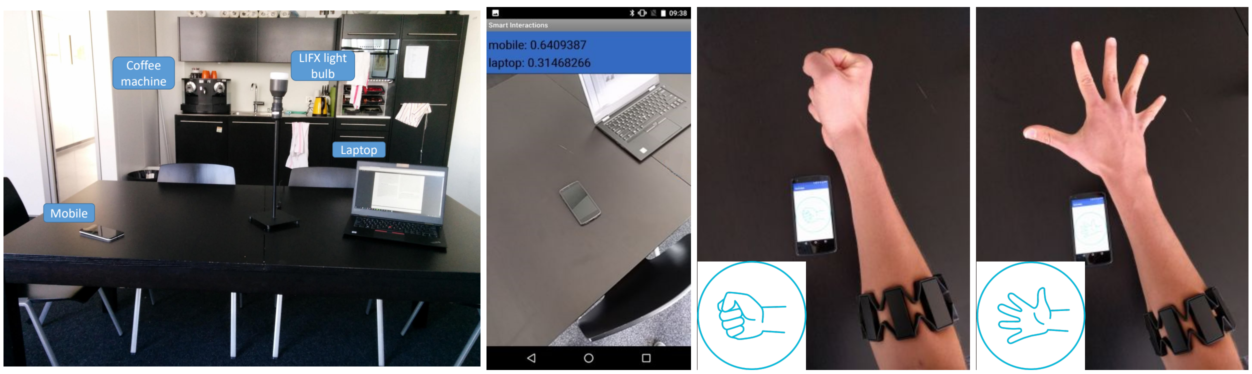
Author(s):
Becker, Vincent; Bâce, Mihai; Sörös, Gábor

Publication date:
2017-09

Permanent link:
https://doi.org/10.3929/ethz-b-000219532

Wearable machine learning for recognizing and controlling smart devices

Vincent Becker, Mihai Bâce, Gábor Sörös
Department of Computer Science
ETH Zurich, Switzerland
{vbecker | mbace | soeroesg}@inf.ethz.ch



(a) (b) (c) (d)

Figure 1: Smart devices in our environment are visually recognized by a smartphone, and their appropriate interaction primitives are mapped to gestures recognized by an electromyography (EMG) band.

ABSTRACT
Augmenting people with wearable technology can enhance their natural sensing, actuation, and communication capabilities. Interaction with smart devices can become easier and less explicit when combining multiple wearables instead of using device-specific apps on a single smartphone. We demonstrate a prototype for smart device control by combining quasi real-time visual device recognition on a smartphone and EMG-based gesture recognition on a Myo armband.

Author Keywords
Smart Object; Deep Learning; Object Recognition; Gesture Control; Wearable Computing; HCI

INTRODUCTION
Appliances in our homes are becoming smarter with improved sensing, actuation, and computing capabilities as well as connectivity to other devices, to the Internet, and to the user's personal devices [4]. We can, for example, control the color and brightness of a smart light bulb, switch channels on a TV, or remotely monitor a heating system using smartphone apps. The advantages of such "outsourced" user interfaces are smaller cost, overcoming the lack of space on devices to display all the features, the ability to control remotely, and simple feature updates via software, to name only a few.

At the same time, potentially a multitude of wearable devices (smartwatch, smartglasses, smartphone, etc.) can augment users with more advanced sensing, actuation, and communication capabilities. The main advantage of wearables compared to other forms of control is that they share the user's egocentric perspective: smartwatches feel how we move, smartglasses see what we see, and smartphones can extract many other contextual cues. Our main interest is how body-worn object recognition and input devices, so far mostly disconnected cyber-physical systems, can seamlessly cooperate in order to better assist the user in everyday tasks.

We present a method to select and control different smart objects by combining off-the-shelf wearables. Objects of interest are automatically recognized in first-person camera images by a convolutional neural network (CNN) running on the user's smartphone. From the range of detected devices, the user can select and control appliances via gestures recognized by a Myo EMG armband[1].

[1] https://www.myo.com

MobileHCI'17 Workshops Sep 04–07, 2017, Vienna, Austria

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

DEVICE RECOGNITION FOR INTERACTION

Object recognition from an egocentric perspective has been applied in previous research to detect the devices a user wants to interact with, in order to make the interaction more natural and intuitive. Mayer et al. use visual object recognition on a smartphone in order to externalise a smart device's capabilities to a graphical representation on the smartphone [5]. Similarly, in Snap-To-It [2] a photo of a device taken with a smartphone is analyzed and the location-wise nearest device matching the picture is selected. A graphical user interface on the smartphone then shows the possible actions of that device. InSight [1] is a system which uses an eyewear-mounted IR laser emitter and IR sensors attached to the smart devices. The laser approximates the user's gaze and makes it possible to detect the devices the user's gaze falls upon.

SYSTEM

Our system consists of two parts: first, a smartphone for obtaining a video stream from an egocentric point of view, for performing the CNN computations, for communication with the smart devices, and propagating the gesture commands and second, a Myo armband for gesture recognition.

Object Detection

To perform the detection of the smart devices, we retrain a pre-trained CNN [3]. The main idea is that the convolutional layers trained on a large dataset such as ImageNet, consisting of millions of images, are strong and robust general-purpose visual feature detectors. Consequently, it may not be necessary to train these layers when adapting to a new set of images and classes, but to only train the top softmax layer.

For retraining, we use an ImageNet pre-trained Inception V1 network[2] (also known as *GoogLeNet* [6]). We aim at detecting devices from five classes: smartphones, laptops, LIFX[3] light bulbs, coffee machines, and sound systems. The network consists of nine inception modules, which contain different convolution operations. As in the *Tensorflow* Android demo[4], featuring a classifier for ImageNet classes, we only use the first three modules (up to inception 4a) and an additional ReLU-layer. We then add a softmax layer on top to adapt the network to our five classes. After stripping the network from all the unused layers, we achieve a model size of about 12 MB, which could easily be shipped within a mobile app. As framework for running the network on the smartphone we use *Tensorflow*. As image dataset, we use around 100 images per class, which we obtain from short self-taken videos and split them into training (80%), validation (10%), and test (10%) set. We train for 200 iterations with a batch size of 50. The whole retraining process only takes about three minutes on an Intel i7-6600U CPU. We reach a test accuracy of 100%. As a test device, we use a Nexus 5X. The inference time for a single image (input resolution 224x224) is about 550ms on average, i.e., an update rate of about 2 Hz. However, we always show the current camera view, so the user does not experience a delay.

[2] https://storage.googleapis.com/download.tensorflow.org/models/inception5h.zip

[3] https://www.lifx.com/

[4] https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android

Gesture Control

We use a Myo armband for gestural input to the recognized smart device. The armband contains eight EMG (electromyography) sensors, which measure the action potential from the muscles in the forearm. These electrical potentials correspond to muscle contractions, hence it is possible to derive the current hand gesture from the EMG measurements. Additionally, there is a nine-axis IMU. The Myo detects a pre-defined set of five gestures, making a *fist*, spreading one's fingers (*fingers_spread*), waving towards (*wave_in*) and away (*wave_out*) from one's body, and double tapping thumb and middle finger (*double_tap*). The Myo communicates to a listening device via Bluetooth. It is also possible to obtain the raw sensor readings, so one can create custom gestures. An advantage of the Myo is that it can be worn over a long period of time and the hand does not have to be in the field of view of any camera or other device.

Interaction with Smart Devices

In this work, we use a smartphone as the central hub. After the Myo armband is connected to the smartphone, the user can initiate the object detection performing the *double_tap* gesture. Subsequently, a camera view is opened and the incoming images are classified. The current labels are displayed as an overlay. The camera on the smartphone could be replaced by a head-mounted camera, e.g., a Google Glass. As soon as the user decides to stop the detection phase by executing the *fist* gesture, a connection to the device currently recognized is set up and the user may begin to interact using the Myo gestures. If the detection results are ambiguous, our app displays these and lets the user choose the correct device using the *wave_in* and *wave_out* gestures for left and right and the *fist* gesture for selection.

APPLICATIONS

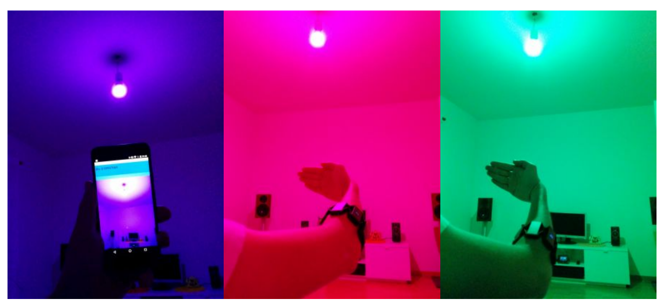


Figure 2: The user interacts with the LIFX light bulb by using the smartphone to detect it and then controlling the light color with the *wave_in* and *wave_out* gestures.

Currently, we have two demo applications. First, connecting to another smartphone via Bluetooth and transmitting the Myo gestures as shown in Figures 1c and 1d and second, connecting to a LIFX light bulb via Wi-Fi and using the *wave_in* and *wave_out* gestures for selecting colors as shown in Figure 2. We demonstrate the described applications and provide a recipe to other researchers of how to setup and retrain a CNN and how to incorporate it in an application for interaction.

REFERENCES

1. R. Boldu, H. Zhang, J. P. F. Cortés, S. Muthukumarana, and S. Nanayakkara. InSight: A Systematic Approach to Create Dynamic Human-controller-interactions. In *Augmented Human 2017*.

2. A. A. de Freitas, M. Nebeling, X. A. Chen, J. Yang, Akshaye S. K. Karthikeyan R., and A. K. Dey. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *CHI 2016*.

3. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *CoRR* (2013). http://arxiv.org/abs/1310.1531

4. F. Mattern. From Smart Devices to Smart Everyday Objects. In *Smart Objects Conference 2003*. 15–16.

5. S. Mayer and G. Sörös. User Interface Beaming - Seamless Interaction with Smart Things using Personal Wearable Computers. In *Body Sensor Networks 2014 Workshops*.

6. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR 2015*.