Low-Power Gesture Recognition Using CNNs and Ambient Lighting

Arne de Beer - a.r.debeer@student.tudelft.nl

Responsible Professor: Qing Wang
Supervisors: Mingkun Yang & Ran Zhu

TU Delft

1. Introduction

Recognising gestures has been of interest for over 30 years [1] [2]. With today's advances in computing power and machine learning (ML) knowledge, we can detect gestures **more efficiently**, **faster** and in **more compact form factors** [3].

This research aims at **using the existing lighting infrastructure** in buildings and ambient lighting to implement a **low-power** system to recognise **several gestures** using three simple OPT101 photodiodes.

This system will use an **ML model** to be able to **recognise and classify** never before seen data. We build upon the foundations laid out by the project performed last year. They researched optimal placement and designed a PCB that is used this year, shown in figure 1.

2. Research Questions

This research concerned three research questions:

- How to efficiently read the signals from three OPT101 photodiodes and process them into 2D data?
- What kind of ML model is best to use for gesture recognition, such as swipe left or right, based on 2D pre-processed data, when inference time must be near real-time and computational processing power is limited, whilst trying to keep the success rate above 75%?
- How to compress a machine learning model to make it real-time on an Arduino Nano 33 BLE? Real-time is considered to be below 200 milliseconds.

3. Data

3.1 2D-Formatted Data

This research focuses on 2D formatted data. The data comes from three photodiodes and can be seen as an image with 3 rows and N columns, where N is the amount of samples taken per photodiode. An example of such an image can be seen in figure 2.

3.2 Pre-Processing & Data Augmentation

Before feeding the data into an ML model, pre-processing is an important step to **increase inference accuracy** by removing noise and unwanted data. Some operations performed in the pre-processing pipeline are:

- Rescaling to a range of [0, 1]
- Normalisation
- Low pass Butterworth filter

The difference between before and after pre-processing the data can be seen in figures 2 and 3.

Furthermore, during training data augmentation techniques were applied to **increase the dataset size and variation**. These consisted of applying random constrast and small random translations.

4. Data Collection

4.1 Participants

Collecting enough data is important for training a **generalisable model**. Each participant was requested to perform each gestures 5 times as naturally as posible. Participants were recruited with no preference for age, gender, or previous experience with technology.

4.2 Dataset

The current dataset consists of 17 left- and 26 right-handed data, with a total of just **over 2300 performed gestures** in various indoor lighting conditions.

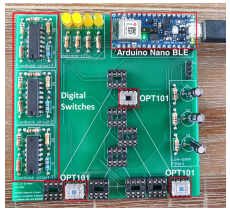

Figure 1: The PCB designed last year.

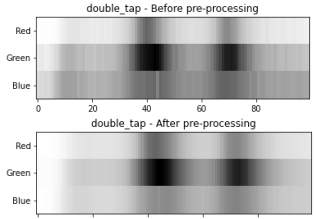

Figure 2: Data formatted as an image. Double tap example before and after pre-processing.

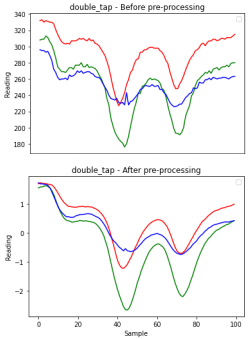

Figure 3: Data formatted in a plot. Double tap example before and after pre-processing.

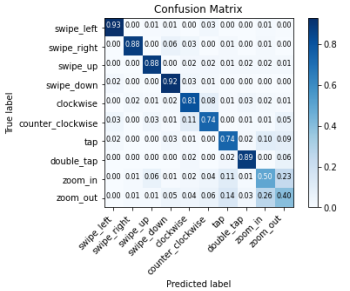

Figure 4: A confusion matrix with all 10 gesture types. It shows how often the model gives a certain gesture type a specific label.

5. System Overview

The resulting system consists of the following parts:

- A proof of concept microcontroller program with a start detection algorithm which constantly checks whether it should input data into the machine learning model.
- Analysis of various pre-processing and data augmentation techniques.
- Analysis of various models and hyperparameters.

6. Results

6.1 Accuracy Testing

The constructed pre-processing pipeline improved the inference accuracy of all explored model architectures from an unusable 10% to 74.5%. Data augmentation added around an additional 2%.

Final results with the best-performing architecture, shown in figure 5, showed an **inference accuracy of 76.8% ± 5.8%**. Various hyper-parameters were experimented with and the final model was trained with 768 epochs, batch size of 256 and reshaped input shape from (100, 3) to (20, 5, 3).

Figure 4 shows how the specific gestures perform. Especially the more trivial swipe gesture perform remarkably well. The confusion between the zoom gestures should definitely still be improved.

6.2 Inference and Processing Latency Testing

All tested models showed an **inference time below 40 milliseconds**. The full pre-processing pipeline only took just a bit over 500 microseconds. This gives a back to back delay well below 50 milliseconds. For real-world usecases an additional delay should probably have to be added in order to prevent duplicate gesture detection.

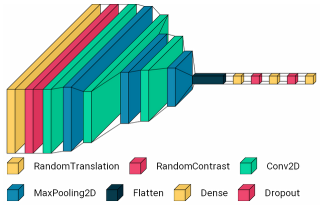

Figure 5: The best-performing model architecture. The first two layers are only used during training for data augmentation.

7. Future Work

- Rethink the zoom gestures: modify or use different gesture.
- Dataset tailored for one specific application.
- Extend the contributed dataset.
- Further experimentation with model architectures and parameters.

[1] D.L. Quam. Gesture recognition with a dataglove. In IEEE Conference on Aerospace and Electronics, pages 755–760 vol.2, 1990.
[2] R. Beale and A.D.N. Edwards. Gestures and neural networks in human-computer interaction. In IEE Colloquium on Neural Nets in Human-Computer Interaction, pages 5/1–5/4, 1990.
[3] Taegeun Yoo, Van Loi Le, Ju Eon Kim, Ngoc Le Ba, Kwang-Hyun Baek, and Tony. T. Kim. A 137-w area-efficient real-time gesture recognition system for smart wearable devices. In 2018 IEEE Asian Solid-State Circuits Conference (A-SSCC), pages 277–280, 2018.