FIT
CMUS

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

NHẬP MÔN LẬP TRÌNH ĐIỀU KHIỂN THIẾT BỊ THÔNG MINH

ĐỀ TÀI

CAT FEEDER WITH SPEECH COMMAND

Sinh viên: La Nhật Hy – MSSV: 18120402

HỌC KỲ 1 / 2021/2022

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2022

Table of Contents

CAT FEEDER WITH SPEECH COMMAND

I. Introduction

Trong thời đại cách mạng công nghiệp 4.0, nhu cầu của con người về sự phục vụ của máy móc cũng phát triển mạnh mẽ theo. Hiện nay, máy móc đã hỗ trợ con người trong rất nhiều lĩnh vực như trong công nghiệp, nông nghiệp, giáo dục, vv. Tuy nhiên trong đời sống hằng ngày, có một số việc tưởng chừng như không thể thay thể bởi con người. Nhưng hiện tại, nhu cầu về sự hổ trợ của máy móc về việc này lại đang hình thành, điển hình là chăm sóc thú cưng.

Trong đồ án môn học này, em sẽ giới thiệu một hệ thống thông minh cho mèo ăn có thể điều khiển được bằng cả nút bấm và đặc biệt là bằng giọng nói. Ưu điểm của hệ thống này là sự tiện lợi về mặt thời gian và không gian. Con người có thể cho thú cưng của mình ăn chỉ bằng việc ra lệnh mà không cần phải tốn thời gian lấy thức ăn cho chúng. Thú cưng cũng có thể tự lấy thức ăn bằng cách chạm vào nút bấm.

Dưới đây là mô tả chi tiết về hệ thống này.

II. Model

1. Data

Mô hình này gồm 3 module: Wake word detection, Intent capture và Intent Excutor. Do đó việc chuẩn bị dữ liệu phải phù hợp với các module này.

1.1 Preparation

Với dữ liệu huấn luyện, mô hình sử dụng tập dữ liệu *speech_commands* version 0.0.2 của Tensorflow. Tập dữ liệu này chứa hơn 100.000 files audio. Mỗi file có độ dài 1 giây. Các files audio này là tập hợp của 20 lệnh. Ví dụ như yes, no, up, down, vv.

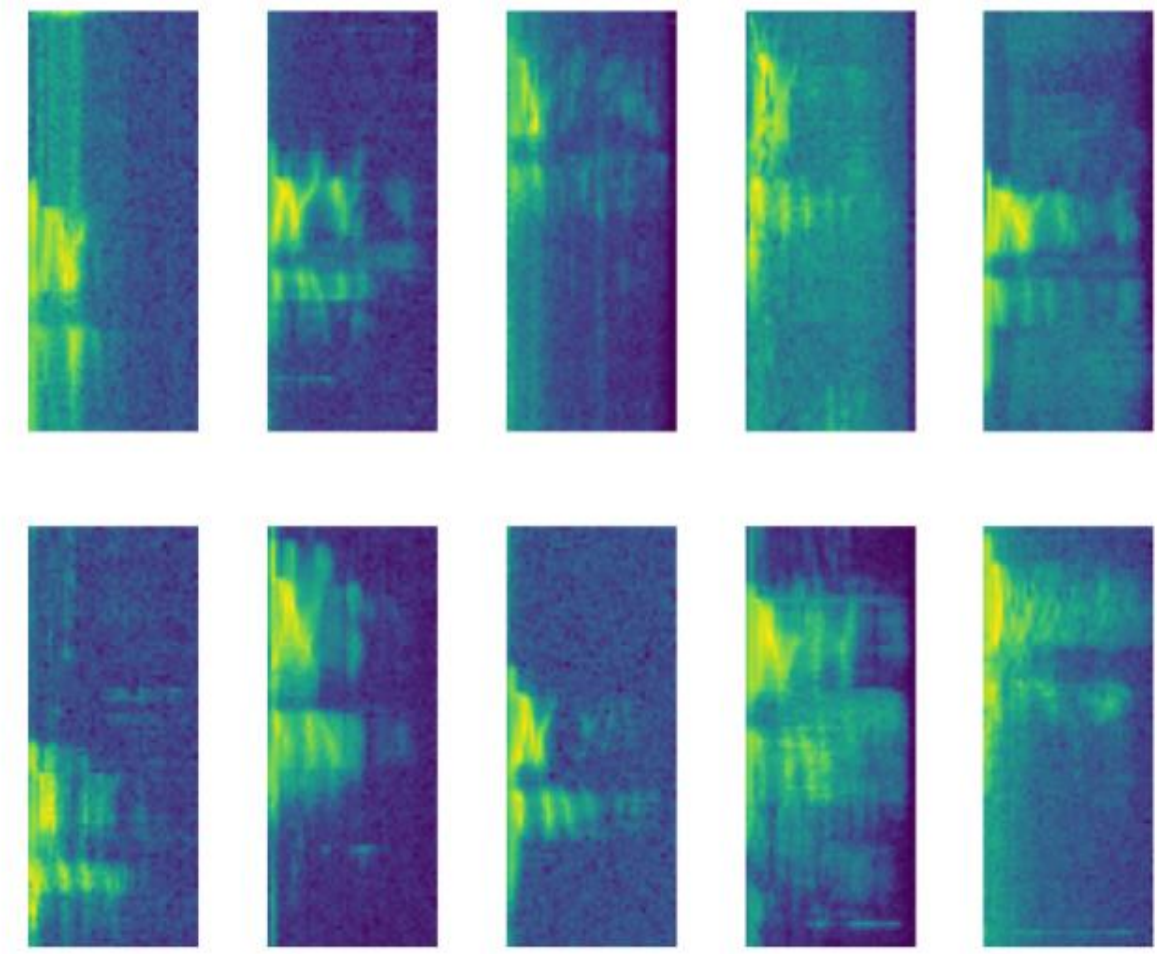
Với module Wake word detection, mô hình sử dụng command phù hợp nhất trong tập dữ liệu là “Marvin”.

Để model có độ chính xác cao hơn, cần thu thập thêm các dữ liệu nhiễu loạn để đưa vào tập dữ liệu.

1.2 Feature Extraction

Các mô hình mạng neuron học sâu hiện nay đều tính toán trên ma trận ảnh. Do đó, để đưa dữ liệu vào huấn luyện mô hình, ta cần chuyển dữ liệu audio đầu vào thành dạng ảnh. Bài toán bây giờ trở thành bài toán nhận dạng trên ảnh. Ý tưởng là chuyển dữ liệu audio thành dạng ảnh spectrogram.

Để lấy được ảnh spectrogram của một file audio, ta cần chia nhỏ file ra thành từng đoạn ngắn. Sau đó, áp dụng phép biến đổi Fourier rời rạc cho các đoạn này. Phép biến đổi cho ra được ảnh đặc trưng tần số của từng đoạn. Ghép các ảnh đặc trưng này lại, ta có ảnh spectrogram của file audio ban đầu.



Ảnh spectrogram của một số mẫu dữ liệu.

2. Training

2.1 Model Architecture

Kiến trúc model đề xuất gồm:

- 2 lớp Conv_2D để rút trích đặc trưng.
- 2 lớp Max pooling để giảm số chiều.
- 1 lớp Flatten.
- 1 lớp Dense kết nối các neuron trước đó và cho ra output.

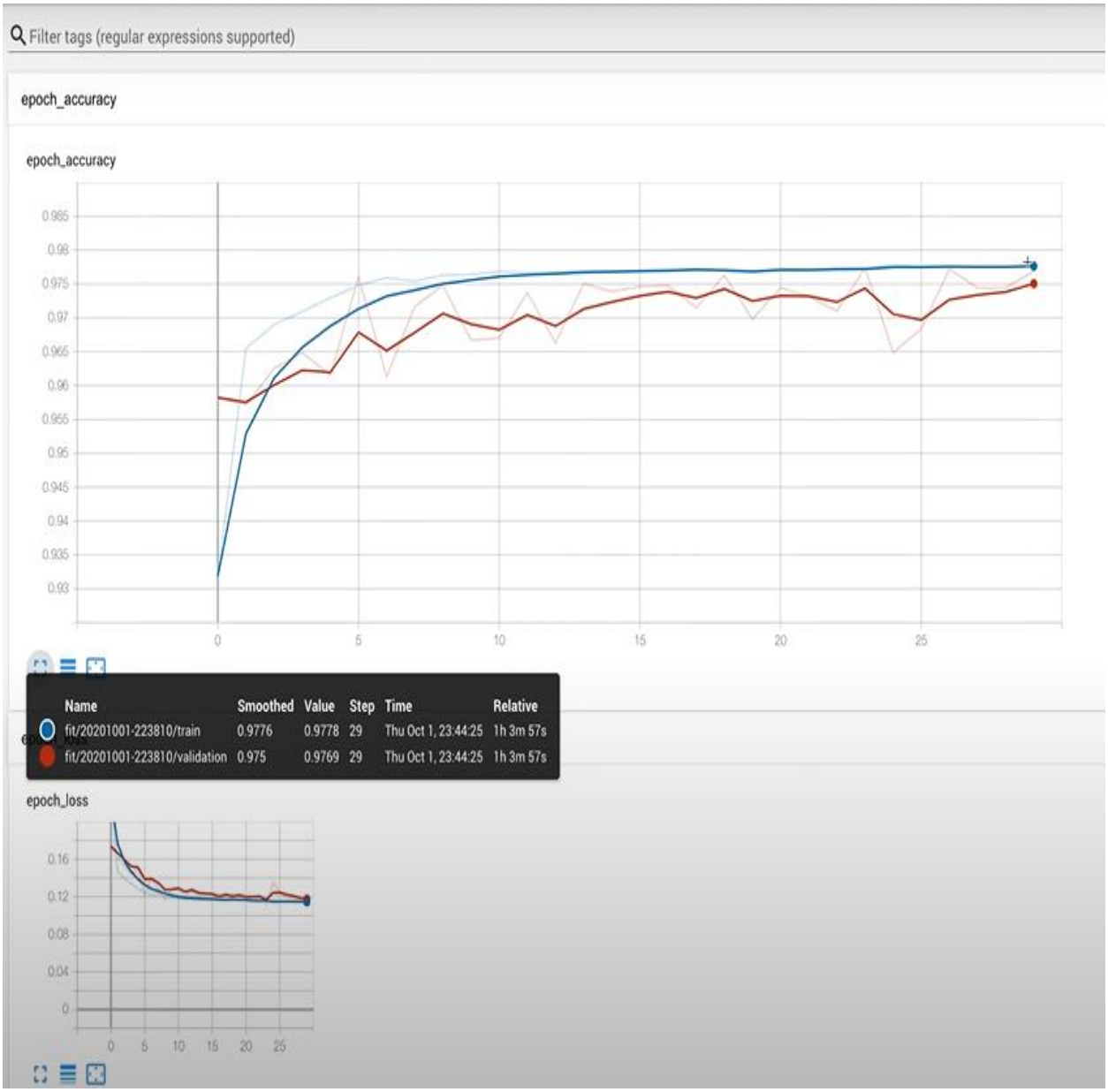
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_layer1 (Conv2D)         (None, 99, 43, 4)         40
_________________________________________________________________
max_pooling1 (MaxPooling2D)  (None, 49, 21, 4)         0
_________________________________________________________________
conv_layer2 (Conv2D)         (None, 49, 21, 4)         148
_________________________________________________________________
max_pooling2 (MaxPooling2D)  (None, 24, 10, 4)         0
_________________________________________________________________
flatten (Flatten)            (None, 960)               0
_________________________________________________________________
dropout (Dropout)            (None, 960)               0
_________________________________________________________________
hidden_layer1 (Dense)        (None, 40)                38440
_________________________________________________________________
output (Dense)               (None, 1)                 41
=================================================================
Total params: 38,669
Trainable params: 38,669
Non-trainable params: 0
_________________________________________________________________
```

Thông tin cài đặt chi tiết cho model.

2.2 Learning Hyper-parameters

- Batch size: 30.
- Learning algorithm: Adam.
- Loss function: Binary cross entropy.

2.3 Evaluation



Biểu đồ đánh giá quá trình train model.

Có thể thấy, hiệu suất huấn luyện (train) gần xấp xỉ hiệu suất đánh giá (validation) với độ chính xác khá cao (~0.975).

Đánh giá confusion matrix: Tỉ lệ false positive là cao hơn so với false negative. Có thể nói mô hình có hiệu suất khá ổn với mức false negative thấp.

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[13616,   427],
       [  743, 11927]], dtype=int32)>
```

3 Deployment

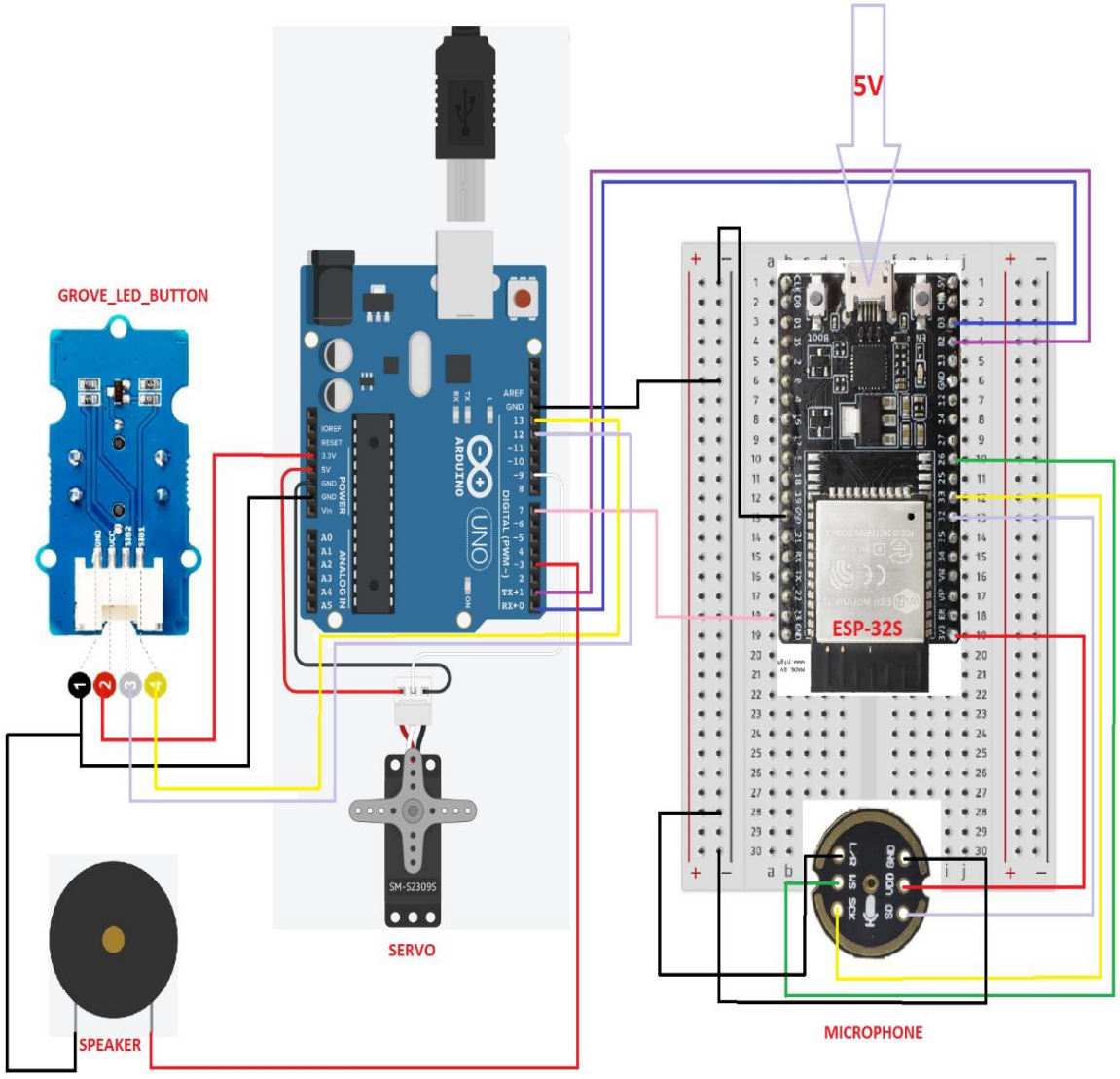
Model sau khi train xong sẽ được định dạng lại để có thể chạy trên thiết bị nhúng (ESP-32). Bằng cách sử dụng Tensorflow Lite, model được định dạng lại dưới dạng file *model.cc*. Tổng kích thước model sau cùng là 44128 bytes.

III. System Operation

1. System description and Schematic

Hệ thống được xây dựng với các linh kiện: Arduino Uno R3, Node-MCU ESP32S, Microphone INMP441 I2S, RC Servo MG996, Adapter, dây dẫn.

Bao gồm 3 module chính: Wake word detection, Intent capture, Intent Excutor. Dưới đây là sơ đồ chi tiết hệ thống.



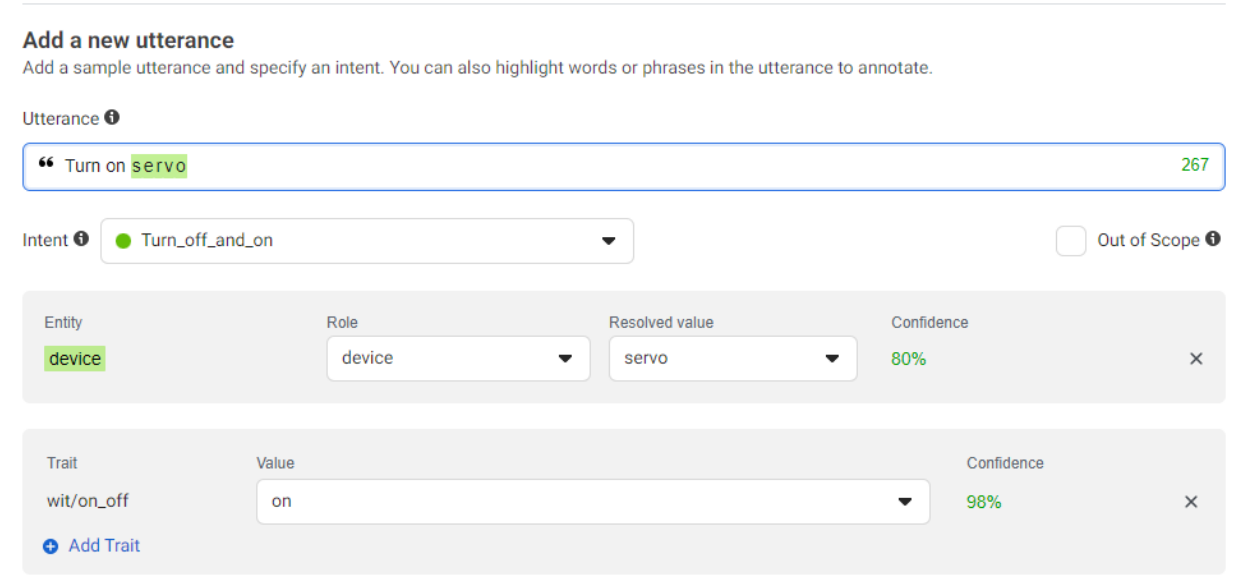
2. Wake word detection

Module này được đảm nhiệm bởi Node-MCU ESP32. Thông qua microphone, thiết bị sẽ liên tục lắng nghe âm thanh xung quanh. Dữ liệu từ microphone được đưa vào model đã được huấn luyện để suy luận keyword. Khi kết quả suy luận từ model vượt qua threshold, module sẽ ngưng suy luận key word và chuyển trạng thái hoạt động cho module kế tiếp.

3. Intent capture

Module này vẫn được đảm nhiệm bởi Node-MCU ESP32. Khi trạng thái hoạt động được kích hoạt, microphone sẽ bắt đầu ghi âm lại các câu lệnh của người dùng trong 3 giây. Kết thúc quá trình ghi âm, dữ liệu vừa được ghi sẽ được gửi lên server *Wit.ai* để tiếp tục nhận dạng. Server sẽ xử lý dữ liệu này và xác định được người dùng đang yêu cầu gì.

Wit.ai là một dịch vụ giúp nhận dạng các câu lệnh mà người dùng khởi tạo. Các câu lệnh này được gọi là Utterances. Mỗi utterance sẽ có các trường Intent (loại ý định), Entity (thực thể) và Trait chứa giá trị quyết định của câu lệnh (ví dụ như on/off). Người có thể tạo các câu lệnh cơ bản, các câu lệnh này sẽ được server lưu lại. Khi có truy vấn từ clients, server sẽ suy luận dựa trên các utterances người dùng đã tạo và gửi về clients thông tin các trường Intent, Entity và Trait. Mỗi trường gửi về sẽ kèm theo một mức điểm nhận dạng nhất định.



Thiết bị ESP32 sẽ thiết lập kết nối và trao đổi thông tin với server *Wit.ai* thông qua **Client Access Token** mà server tạo.

Sau khi server gửi thông tin về client là thiết bị ESP32, thông tin sẽ được xử lý và so sánh điểm nhận dạng (confidence score) với threshold. Nếu điểm vượt ngưỡng, tín hiệu kích hoạt servo sẽ được gửi đi bằng cách set một chân PIN với mức điện áp HIGH.

4. Intent Excutor

Module này bao gồm board Arduino, Servo và Speaker. Arduino với nhiệm vụ chính là nhận tín hiệu một chân PIN từ ESP32 của module trước đó. Nếu chân PIN có mức điện áp HIGH, speaker sẽ phát ra âm thanh để thu hút thú cưng. Sau đó, servo sẽ được kích hoạt để xoay khay chứa thức ăn. Thức ăn sau đó sẽ được đổ ra cho thú cưng ăn. Cuối cùng, servo quay về vị trí ban đầu. Thức ăn chỉ được đổ ra với một lượng nhất định.

Sau khi xoay servo xong, board Arduino sẽ gửi tín hiệu *flag = 0* thông qua kết nối Serial về cho ESP32. Thiết bị ESP32 khi nhận được *flag* sẽ set điện áp ở chân PIN về mức LOW. Điều này đảm bảo câu lệnh chỉ được thực thi một lần duy nhất.

Ngoài ra, sevo có thể được xoay bằng nút bấm. Khi phím được ấn, điện áp tín hiệu của nút bấm là HIGH. Board Arduino khi nhận được điện áp HIGH ở chân PIN của nút bấm sẽ kích hoạt xoay servo. Thức ăn sẽ được đổ ra cho thú cưng.

IV. Demo

References

[1] atomic14, "GitHub -," 04 10 2020. [Online]. Available: https://github.com/atomic14/diy-alexa.

[2] A. Ulitin, "hackster.io - DIY Arduino Cat Feeder," 26 08 2016. [Online]. Available: https://www.hackster.io/momwillbeproud/diy-arduino-cat-feeder-1c4c7f.