



รายงาน

เรื่อง สร้างระบบเพื่อจำแนกเสียงเป็น 'ใช่' หรือ 'ไม่ใช่'

โดยขึ้นอยู่กับพลังงาน

โดย

1. นายชภัทรภูมิ นพศรี 640910670
2. นายธฤตวัน จีวาคม 640911030
3. นางสาวพรหมวศ อภิรักษ์รัตนกุล 640911046
4. นายคมชาญ ตริพงศ์พิพัฒน์ 640911145

เสนอ

รองศาสตราจารย์ดร. ชูเกียรติ สอดศรี

รายงานฉบับนี้เป็นส่วนหนึ่งของการเรียนวิชา 618362 การประมวลสัญญาณดิจิทัล
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ มหาวิทยาลัยศิลปากร

ภาคการเรียนที่ 2 ปีการศึกษา 2566

บทนำ

นี่คือการแก้ไขและปรับปรุงข้อความที่มาจากการจำแนกเสียงเป็น "ใช่" หรือ "ไม่ใช่" ด้วย Zero Crossing Rate (ZCR) ความสำคัญการจำแนกเสียงเป็นกลุ่ม "ใช่" หรือ "ไม่ใช่" เป็นกระบวนการที่สำคัญในการประยุกต์ใช้ในหลากหลายสถานการณ์ เช่น การระบุคำพูด การจำแนกเสียงเป็น "ใช่" หรือ "ไม่ใช่" สามารถใช้ในการระบุคำพูดที่เป็นเสียงการตอบ "ใช่" หรือ "ไม่ใช่" ในระบบแชทบอตหรือระบบสนทนาอัตโนมัติการตรวจจับเหตุการณ์ การจำแนกเสียงเป็น "ใช่" หรือ "ไม่ใช่" สามารถใช้ในการตรวจจับเหตุการณ์ เช่น เสียงการทำงานของเครื่องจักรหรือเสียงอัตราการเคลื่อนที่การควบคุม การจำแนกเสียงสามารถใช้ในการควบคุมระบบ เช่น การควบคุมการเปิด-ปิดอุปกรณ์โฮมออโตเมชันโดยใช้เสียงคำสั่งการวิเคราะห์สื่อ การจำแนกเสียงสามารถช่วยในการวิเคราะห์สื่อ เช่น การจำแนกเสียงความรู้สึกในเสียงเพลงหรือการวิเคราะห์สัญญาณเสียงที่เกี่ยวข้องกับความสนใจในสื่อต่างๆการใช้ ZCR เป็นลักษณะที่สำคัญในการแยกแยะเสียง เนื่องจากมันสามารถช่วยให้เราสามารถตรวจจับลักษณะพื้นฐานของเสียงได้โดยไม่ต้องใช้ทรัพยากรมากของการคำนวณและมีประสิทธิภาพในการจำแนกเสียงในหลากหลายสถานการณ์ผลลัพธ์ที่ได้ยังเป็นไปตามแนวโน้มในการพัฒนาเทคโนโลยีเสียงและการประมวลผลเสียงในอนาคตได้ด้วยครับ

ขอบคุณครับ

เนื้อหา

Zero Crossing Rate (ZCR) หมายถึง อัตราการเริ่มต้นผ่านศูนย์ของสัญญาณเสียง เป็นค่าที่บ่งชี้ถึงการเปลี่ยนแปลงของเสียงในด้านของเขตต่าง ๆ ของขนาดของออกเหตุการณ์ เป็นสิ่งสำคัญในการวิเคราะห์และประมวลผลเสียง เช่นในการระบุคุณลักษณะของเสียงและการจำแนกเสียง อัตราการเริ่มต้นที่สูงจะแสดงถึงการเปลี่ยนแปลงของสัญญาณเสียงมากขึ้น ส่วนค่าที่ต่ำจะบ่งบอกถึงสัญญาณที่เปลี่ยนแปลงน้อยลง การคำนวณ ZCR มักทำได้โดยการนับจำนวนครั้งที่สัญญาณเสียงเปลี่ยนเครื่องหมายเฉื่อย (crosses) ของแกนศูนย์ แล้วนำไปหารด้วยจำนวนตัวอย่างทั้งหมดในช่วงเวลาหรือตัวอย่างที่กำหนดไว้

ขั้นตอน

นี่คืออธิบายขั้นตอนการสร้างระบบที่สามารถจำแนกเสียงเป็น 'ใช่' หรือ 'ไม่ใช่' โดยใช้ Zero Crossing Rate (ZCR) ในภาษาไทย

1. **เก็บข้อมูล** ก่อนอื่นคุณต้องมีชุดข้อมูลของตัวอย่างเสียงที่มีป้ายชื่อว่า 'ใช่' หรือ 'ไม่ใช่' คุณสามารถบันทึกตัวอย่างเองหรือใช้ชุดข้อมูลที่มีอยู่แล้วก็ได้
2. **การแยกข้อมูลลักษณะ** แยกข้อมูลลักษณะจากตัวอย่างเสียง หนึ่งในลักษณะที่คุณสามารถแยกได้คือ Zero Crossing Rate (ZCR) หรือ ZCR เป็นการวัดว่าเสียงข้างในถูกพลังงานที่เท่ากับศูนย์มีอัตราการเปลี่ยนแปลงเท่าใด ค่า ZCR สูงมักหมายถึงสัญญาณความถี่สูงหรือสัญญาณรบกวน
3. **กวนการประมวลผลก่อน** ปรับค่าตัวอย่างเสียงและแบ่งส่วนย่อยเพื่อวิเคราะห์
4. **โมเดลเรียนรู้ของเครื่อง** เลือกอัลกอริทึมการเรียนรู้ของเครื่องเพื่อฝึกโมเดลการจำแนกของคุณ ตัวเลือกยอดนิยมรวมถึง Support Vector Machines (SVM) Random Forests หรือ Convolutional Neural Networks (CNNs)
5. **การฝึก** ฝึกโมเดลของคุณด้วยลักษณะที่แยกไว้จากตัวอย่างเสียงที่มีป้ายชื่อใช้เทคนิค เช่น cross-validation เพื่อปรับพารามิเตอร์และป้องกันการเกิด overfitting
6. **การประเมิน** ประเมินประสิทธิภาพของโมเดลของคุณโดยใช้เมตริกเช่นความแม่นยำ ความแม่นยำ ความจำเพาะ และคะแนน F1 บนชุดตรวจสอบแยกต่างหาก
7. **การใช้งาน** เมื่อพอใจกับประสิทธิภาพของโมเดลคุณสามารถใช้งานในแอปพลิเคชันจริงๆ เพื่อจำแนกเสียงที่เข้ามาเป็น 'ใช่' หรือ 'ไม่ใช่' โดยดูจากค่า ZCR ของเสียงที่เข้ามาใหม่

1.Code ที่ใช้ในการอัดเสียง

```
system_energy_ZCR.m x audio_recording.m x +
1   recObj = audiorecorder(44000, 24, 1);
2   for i=1:5
3       fprintf('Start speaking for audio #%d\n',i)
4       recordblocking(recObj, 2);
5       fprintf('Audio #%d ended\n',i)
6
7       y = getaudiodata(recObj);
8       y = y - mean(y);
9       file_name = sprintf('C:/Users/plurk/Desktop/SS pp/test/yes/yes%d.wav',i);
10      audiowrite(file_name, y, recObj.SampleRate);
11  end
```

2.เรียนรู้จากเสียงที่อัดไว้และวิเคราะห์เสียงที่อัดไปใหม่ว่าเสียงพูดเป็น yes หรือ no

```
system_energy_ZCR.m x audio_recording.m x +
1   training_files_yes = dir('C:\Users\plurk\Desktop\SS pp\train\yes\*.wav');
2   testing_files_yes = dir('C:\Users\plurk\Desktop\SS pp\test\yes\*.wav');
3   training_files_no = dir('C:\Users\plurk\Desktop\SS pp\train\no\*.wav');
4   testing_files_no = dir('C:\Users\plurk\Desktop\SS pp\test\no\*.wav');
5
6   data_yes = [];
7   for i = 1:length(training_files_yes)
8       file_path = strcat(training_files_yes(i).folder, '\', training_files_yes(i).name);
9       [y,fs] = audioread(file_path);
10
11       ZCR_yes1 = mean(abs(diff(sign(y(1:floor(end/3))))))./2;
12       ZCR_yes2 = mean(abs(diff(sign(y(floor(end/3):floor(end*2/3))))))./2;
13       ZCR_yes3 = mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
14
15       energy = sum(y.^2);
16       ZCR_yes = [ZCR_yes1 ZCR_yes2 ZCR_yes3 energy];
17       data_yes = [data_yes ; ZCR_yes];
18   end
19   ZCR_yes=mean(data_yes);
20   fprintf('The ZCR of yes is \n');
21   disp(ZCR_yes);
22
23   |
24   data_no = [];
25   for i = 1:length(training_files_no)
26       file_path = strcat(training_files_no(i).folder, '\', training_files_no(i).name);
27       [y,fs] = audioread(file_path);
```

```

29
30 ZCR_no1 = mean(abs(diff(sign(y(1:floor(end/3))))))./2;
31 ZCR_no2 = mean(abs(diff(sign(y(floor(end/3):floor (end*2/3))))))./2;
32 ZCR_no3 = mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
33
34 energy = sum(y.^2);
35
36 ZCR_no = [ZCR_no1 ZCR_no2 ZCR_no3 energy];
37
38 data_no = [data_no ;ZCR_no];
39 end
40 ZCR_no=mean(data_no);
41 fprintf('The ZCR of no is \n');
42 disp(ZCR_no);
43
44
45 for i = 1:length(testing_files_yes)
46 file_path = strcat(testing_files_yes(i).folder,'\',testing_files_yes(i).name);
47 [y,fs] = audioread(file_path);
48 ZCR_yes1 = mean(abs(diff(sign(y(1:floor(end/3))))))./2;
49 ZCR_yes2 = mean(abs(diff(sign(y(floor(end/3):floor (end*2/3))))))./2;
50 ZCR_yes3 = mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
51 energy = sum(y.^2);
52
53 y_ZCR = [ZCR_yes1 ZCR_yes2 ZCR_yes3 energy];
54 if(pdist([y_ZCR;ZCR_yes],'cosine') < pdist([y_ZCR;ZCR_no],'cosine'))
55     fprintf('Test file [yes] #%d classified as yes \n',i);
56 else
57     fprintf('Test file [yes] #%d classified as no \n',i);
58 end
59 end
60
61 for i = 1:length(testing_files_no)
62 file_path = strcat(testing_files_no(i).folder,'\',testing_files_no(i).name);
63 [y,fs] = audioread(file_path);
64 ZCR_no1 = mean(abs(diff(sign(y(1:floor(end/3))))))./2;
65 ZCR_no2 = mean(abs(diff(sign(y(floor(end/3):floor (end*2/3))))))./2;
66 ZCR_no3 = mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
67 energy = sum(y.^2);
68
69 y_ZCR = [ZCR_no1 ZCR_no2 ZCR_no3 energy];
70 if(pdist([y_ZCR;ZCR_yes],'cosine') < pdist([y_ZCR;ZCR_no],'cosine'))
71     fprintf('Test file [no] #%d classified as yes \n',i);
72 else
73     fprintf('Test file [no] #%d classified as no \n',i);
74 end
75 end

```

3.ผลการรัน

```
>> system_energy_ZCR
The ZCR of yes is
1.0e+03 *

0.0000    0.0000    0.0000    1.1849

The ZCR of no is
0.0115    0.0347    0.0269    813.6594

Test file [yes] #1 classified as yes
Test file [yes] #2 classified as yes
Test file [yes] #3 classified as yes
Test file [yes] #4 classified as yes
Test file [yes] #5 classified as yes
Test file [no] #1 classified as no
Test file [no] #2 classified as no
Test file [no] #3 classified as no
Test file [no] #4 classified as no
Test file [no] #5 classified as no
>>
```

สรุปผลการทดลอง ใช่ 5 ครั้ง ‘ไม่ใช่’ 5 ครั้งการวิเคราะห์:จากผลการทดลองทั้งหมด ทำให้เห็นว่ามี ใช่ 5 ครั้ง และ ‘ไม่ใช่’ 5 ครั้ง ซึ่งผลลัพธ์ที่ได้มีการแบ่งแยกอย่างเท่าเทียมกันระหว่าง ใช่ และ ‘ไม่ใช่’การทดลองนี้อาจแสดงถึงความสามารถในการจำแนกเสียงของระบบในเงื่อนไขการทดลองนี้ อย่างไรก็ตามควรพิจารณาเพิ่มเติมเกี่ยวกับความแม่นยำของระบบโดยใช้การวิเคราะห์เชิงลึกเพิ่มเติม เช่น การตรวจสอบการคาดการณ์ที่ผิดพลาด หรือการใช้ชุดข้อมูลทดสอบที่หลากหลายเพื่อประเมินประสิทธิภาพของระบบในสถานการณ์ที่หลากหลายได้โดยละเอียด

เอกสารอ้างอิง

- Ahmad DarKhalil. (2017, 16 November). yes-no-recognition-using-matlab. GitHub. GitHub
- AhmadDarKhalil/yes-no-recognition-using-matlab: These files for DSP system to identify 'yes' and 'no' using matlab and signal processing