

Module 13

Sound
Classifier

ml5.js



Module 13: Sound Classification with ml5.js

Sketch 13.1 the preload

Sketch 13.2 classifying

Sketch 13.3 got the results

Sketch 13.4 display

Sketch 13.5 a moving ball

Sketch 13.6 a splash of colour and text



Introduction to module 13 Sound Classifier

This example demonstrates Sound classification using SpeechCommands18w pretrained model. We will look at the basic model and see how good it is and then use some of those commands to move an object around. There are 18 words and they are as follows:

Zero
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
Up
Down
Left
Right
Go
Stop
Yes
No

You will need to allow access to the microphone on your computer (assuming there is one).



Sketch 13.1 the preload

We will preload the pretrained model.

```
let classifier

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
}
```



Sketch 13.2 classifying

We want to start the pretrained model classifying the sounds/words

```
let classifier

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
  classifier.classifyStart(gotResults)
}

function draw()
{
  background(220)
}
```



Sketch 13.3 got the results

Once we have the results we need the callback function `gotResults()` and predict the label which will appear as the top (best) confidence score - `index[0]`. We will add the console log temporarily to see the results.

```
let classifier
let predictedWord = ""

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
  classifier.classifyStart(gotResults)
}

function draw()
{
  background(220)
}

function gotResults(results)
{
  predictedWord = results[0].label
  console.log(predictedWord)
}
```

Notes

It will pick up every sound and try to classify it. The output also depends on what data it is trained on, whose voices and what accents (American or a mixture). Another factor may be the sensitivity of your built in microphone.

We get the right answer but also lots of wrong ones



```
Console Clear ▼  
nine  
③ six  
up  
two  
six  
up  
eight  
six  
② up  
go  
left  
③ three  
six  
eight
```



Sketch 13.4 display

We can remove the console log and display the result on the canvas

```
let classifier
let predictedWord = ""

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
  classifier.classifyStart(gotResults)
}

function draw()
{
  background(220)
  if (predictedWord !== "")
  {
    fill(0, 200, 0)
    textAlign(CENTER, CENTER)
    textSize(64)
    text(predictedWord, width/2, height/2)
  }
}

function gotResults(results)
{
```



```
predictedWord = results[0].label  
}
```

Displayed on the canvas





Sketch 13.5 a moving ball

Removing the text result in draw, we will move the ball around the canvas with our voice. Added the edges unction so that if it goes off the edge you don't lose it entirely.

```
let classifier
let predictedWord = ""
let x = 200
let y = 200

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
  classifier.classifyStart(gotResults)
}

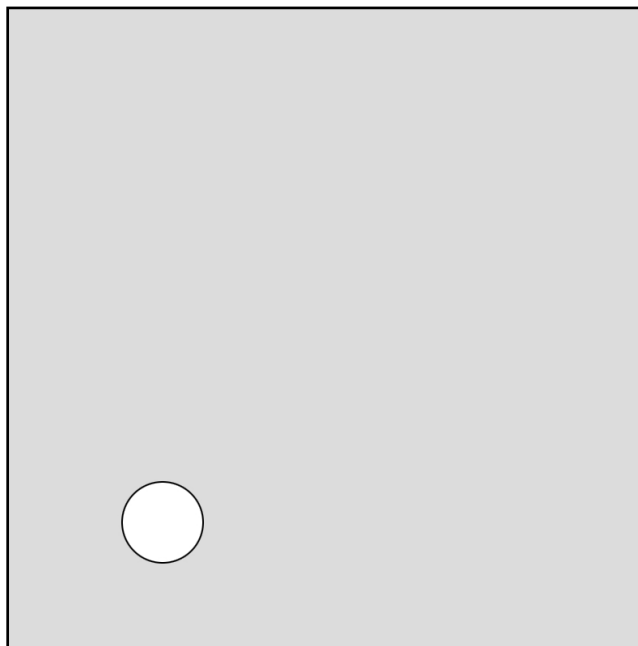
function draw()
{
  background(220, 0, 200)
  fill(200, 200, 10)
  noStroke()
  if (predictedWord == "stop")
  {
    circle(x, y, 50)
    x = x
    y = y
  }
}
```

```
}
else if (predictedWord == "up")
{
    circle(x, y, 50)
    y -= 5
}
else if (predictedWord == "down")
{
    circle(x, y, 50)
    y += 5
}
else if (predictedWord == "left")
{
    circle(x, y, 50)
    x -= 5
}
else if (predictedWord == "right")
{
    circle(x, y, 50)
    x += 5
}
else
{
    circle(x, y, 50)
}
edges()
}

function gotResults(results)
{
    predictedWord = results[0].label
}
}
```

```
function edges()  
{  
  if (x > width)  
  {  
    x = 0  
  }  
  if (x < 0)  
  {  
    x = width  
  }  
  if (y > height)  
  {  
    y = 0  
  }  
  if (y < 0)  
  {  
    y = height  
  }  
}
```

The basic premise





Sketch 13.6 a splash of colour and text

Adding a splash of colour and the text returns

```
let classifier
let predictedWord = ""
let x = 200
let y = 200

function preload()
{
  let options = {probabilityThreshold: 0.7}
  classifier = ml5.soundClassifier("SpeechCommands18w")
}

function setup()
{
  createCanvas(400, 400)
  classifier.classifyStart(gotResults)
}

function draw()
{
  background(220, 0, 200)
  fill(200, 200, 10)
  noStroke()
  if (predictedWord == "stop")
  {
    circle(x, y, 50)
    x = x
    y = y
  }
  else if (predictedWord == "up")
```

```

{
  circle(x, y, 50)
  y -= 5
}
else if (predictedWord == "down")
{
  circle(x, y, 50)
  y += 5
}
else if (predictedWord == "left")
{
  circle(x, y, 50)
  x -= 5
}
else if (predictedWord == "right")
{
  circle(x, y, 50)
  x += 5
}
else
{
  circle(x, y, 50)
}
edges()
if (predictedWord != "")
{
  fill(0)
  textAlign(CENTER, CENTER)
  textSize(64)
  text(predictedWord, width/2, height/2)
}
}

```

```
function gotResults(results)
{
  predictedWord = results[0].label
}

function edges()
{
  if (x > width)
  {
    x = 0
  }
  if (x < 0)
  {
    x = width
  }
  if (y > height)
  {
    y = 0
  }
  if (y < 0)
  {
    y = height
  }
}
```

An interesting experiment

