Project 181

செயற்றிட்டம் 181

DP EDUCATION

Coding School

AI and Machine Learning

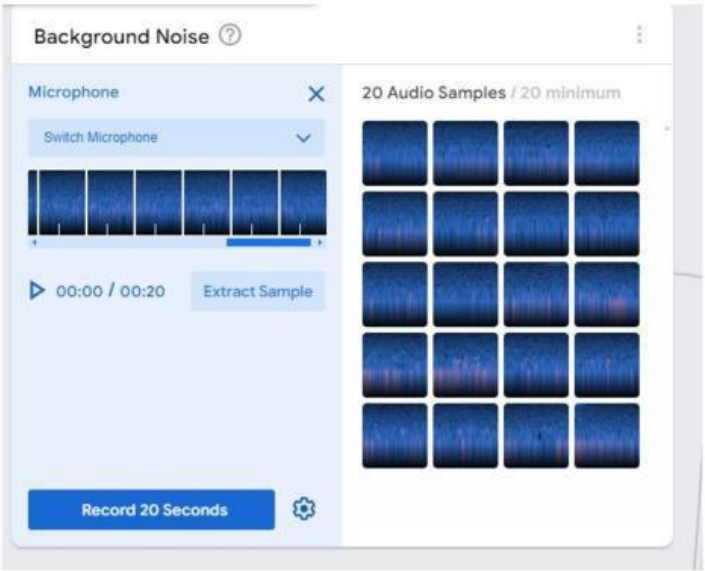
இணையப் பக்கத்தை பார்க்கவும்

ஆரம்பிக்கவும்

LIVEWORKSHEETS

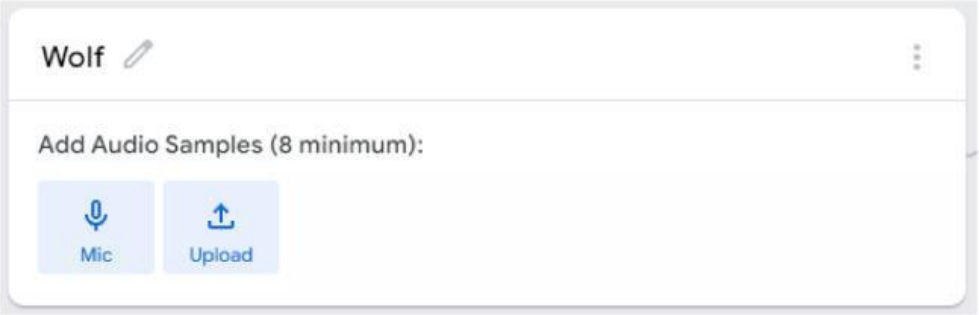
Teachable Machine Audio Project பயன்படுத்தி மிருகங்களின் ஒலியை அடையாளம் காணும் app ஒன்றை உருவாக்குவோம்.

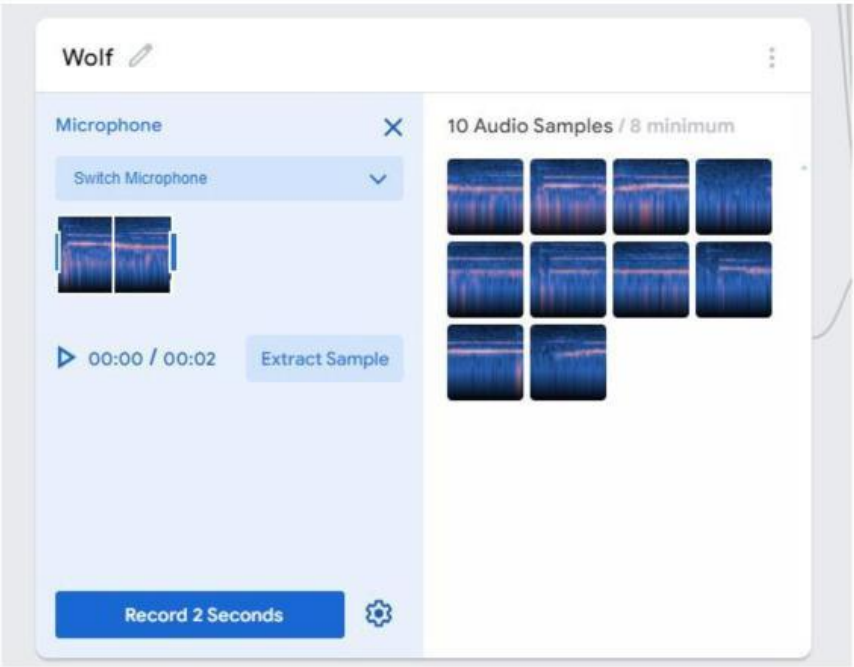
- முதலில் https://teachablemachine.withgoogle.com website இற்கு சென்று get started மூலம் train பண்ணுவதை ஆரம்பிக்கவும். அந்த Audio project இற்குள் சென்று Model ஐ train பண்ணுவதற்கு ஆரம்பிக்கவும் .
- முதலில் background noise இற்குரிய Samples add பண்ணிக் கொள்வோம். அதற்காக mic மூலம் வினாடி 20 இற்கு அதிகமான background sound பகுதி ஒன்றை record பண்ணிக் கொள்க.



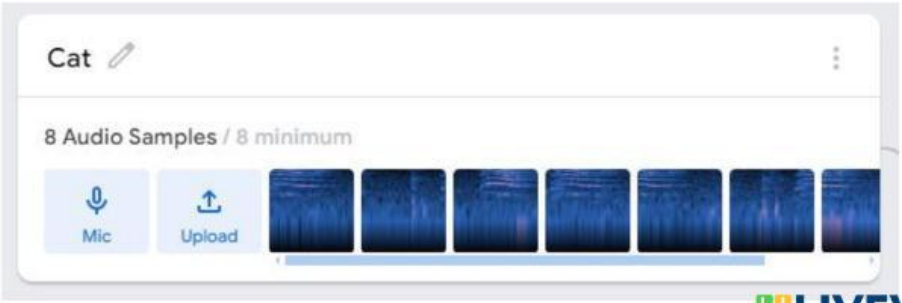
LIVEWORKSHEETS

- கீழே உள்ளவாறு model இல் ஒவ்வொரு மிருகங்களினதும் ஒலிக்கு உரிய samples add பண்ணிக்கொள்ள class உருவாக்கிக் கொள்க.
- உங்களுக்கு ஆரம்பிக்கவும் என தரப்பட்டுள்ள glitch project இன் assets இற்குள் உள்ள மிருகங்களின் ஒலியை பயன்படுத்தவும்.

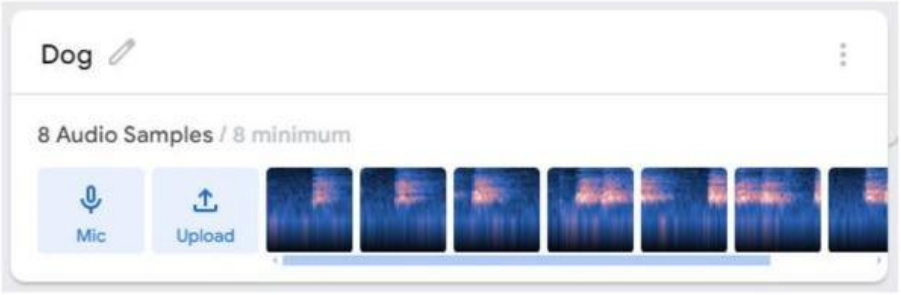


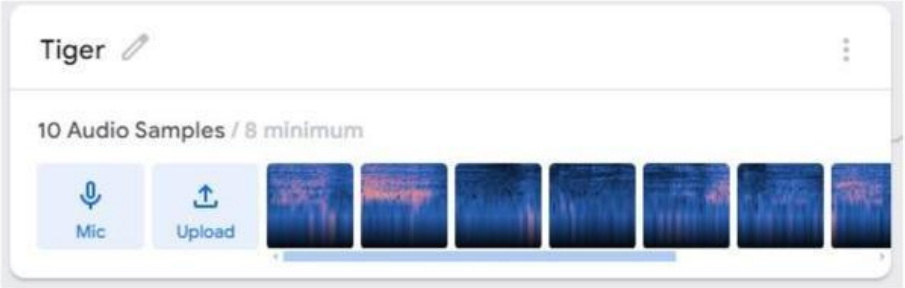


- தலா 2 வினாடிகள் sample record பண்ணி extract sample button இன் மூலம் sample 8 இற்கு அதிகமான அளவு add பண்ணிக் கொள்க.

- 

LIVEWORKSHEETS

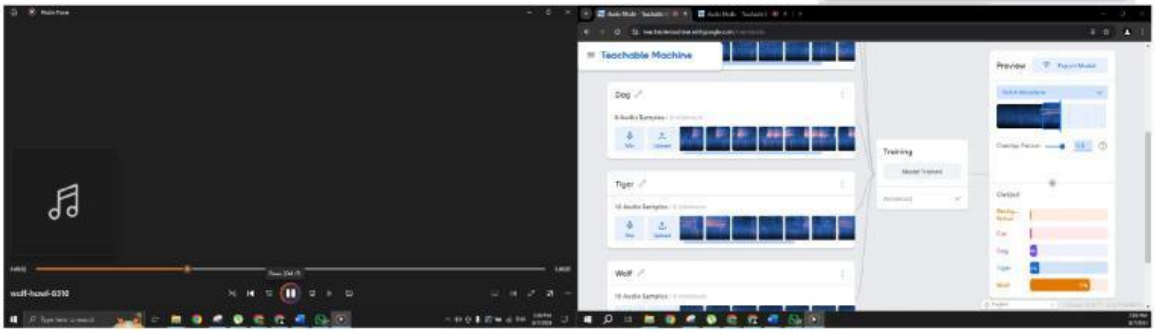




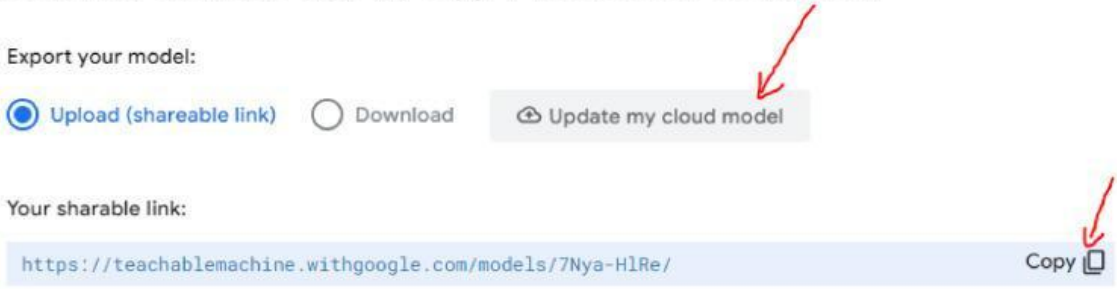
- மேலுள்ளவாறு எல்லா samples classes இற்கு add பண்ணிய பின் train model button மூலம் model ஐ train பண்ணிக் கொள்க.
- Model ஐ train பண்ணிய பின் teachable machine இற்குள்ளேயே உங்களுக்கு Test பண்ணுவதற்கு முடியும். மிருகம் ஒன்றின் ஒலி ஒன்று play ஆகாத போது background noise output என predict ஆக வேண்டும்.

LIVEWORKSHEETS

❖ பின்வரும் மிருகம் ஒன்றின் ஒலி ஒன்று play ஆகும்போது அது Predict ஆக வேண்டும்.



❖ அவ்வாறு test பண்ணிய பின் model ஐ export பண்ணிக் கொள்க. Sharable link ஐ copy பண்ணிக் கொள்க.



❖ இவ்வாறு model ஐ train பண்ணி அதன் link பெற்றுக் கொள்ளப்பட்ட பின் web ஐ உருவாக்குவதற்கு ஆரம்பிப்போம்.

LIVEWORKSHEETS

- உங்களுக்கு ஆரம்பிப்பதற்கு பெற்று தரப்பட்டுள்ள glitch project மூலம் code பண்ணுவதை ஆரம்பிக்கவும்.
- Index file இற்குள் பின்வருமாறு elements உருவாக்கிக் கொள்க.

```
<body>
  <h1>Teachable Machine Audio Model</h1>
  <button type="button" onclick="init()">Start</button>
  <div id="label-container"></div>
```

- அதன் பின் body tag இற்குள் பின்வருமாறு script tags உருவாக்கிக் கொள்க.

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/speech-commands@0.4.0/dist/speech-commands.min.js"></script>
```

- மேலும் script tag ஒன்றை open பண்ணி பின்வருமாறு functions உருவாக்கிக் கொள்க.

```
<script type="text/javascript">

</script>
```

- அந்த script tag இற்குள் முதலில் variable பின்வருமாறு உருவாக்கி நீர் copy பண்ணிய sharable link இற்குள் களஞ்சியபடுத்தவும்.

```
const URL = "https://teachablemachine.withgoogle.com/models/7Nya-HlRe/";
```

- அதன் பின் script tag இற்குள் creatModel function பின்வருமாறுஉருவாக்கிக் கொள்க.

```
async function createModel() {
    const checkpointURL = URL + "model.json"; // model topology
    const metadataURL = URL + "metadata.json"; // model metadata

    const recognizer = speechCommands.create(
        "BROWSER_FFT", // fourier transform type, not useful to change
        undefined, // speech commands vocabulary feature, not useful for your models
        checkpointURL,
        metadataURL);

    // check that model and metadata are loaded via HTTPS requests.
    await recognizer.ensureModelLoaded();

    return recognizer;
}
```

LIVEWORKSHEETS

❖ கடைசியாக init function ஐ பின்வருமாறு உருவாக்கிக் கொள்க.

```
async function init() {
    const recognizer = await createModel();
    const classLabels = recognizer.wordLabels(); // get class labels
    const labelContainer = document.getElementById("label-container");
    for (let i = 0; i < classLabels.length; i++) {
        labelContainer.appendChild(document.createElement("div"));
    }
    recognizer.listen(result => {
        const scores = result.scores;

        for (let i = 0; i < classLabels.length; i++) {
            const classPrediction = classLabels[i] + ": " + result.scores[i].toFixed(2);
            labelContainer.childNodes[i].innerHTML = classPrediction;
        }
    }, {
        includeSpectrogram: true,
        probabilityThreshold: 0.75,
        invokeCallbackOnNoiseAndUnknown: true,
        overlapFactor: 0.50
    });
}
```

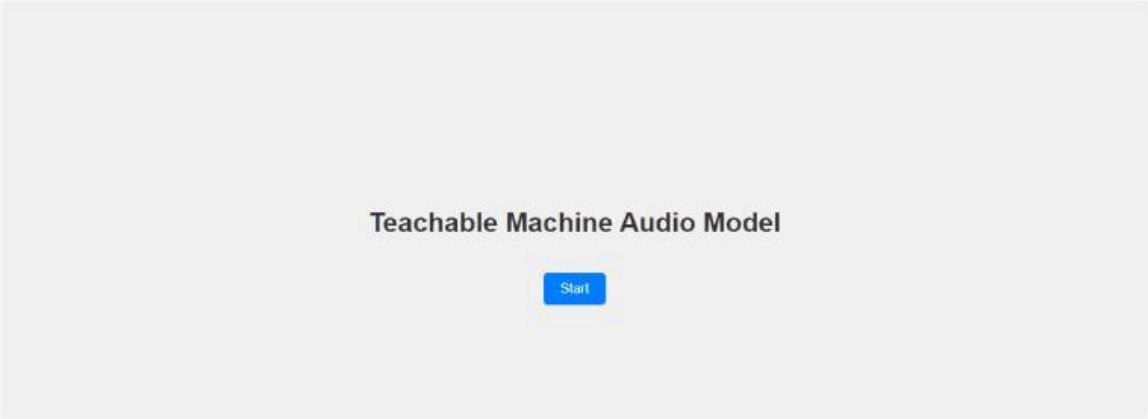
❖ Style file இற்குள் பின்வருமாறு css உருவாக்கிக் கொண்டு web page உருவாக்கிக் கொள்வோம்.

```
body {
        font-family: Arial, sans-serif;
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
        height: 100vh;
        margin: 0;
        background-color: #f0f0f0;
    }
```

LIVEWORKSHEETS

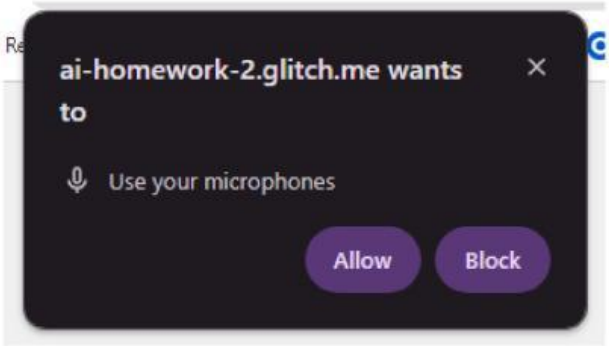
```css
h1 {
    color: #333;
}
button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-top: 20px;
}
button:hover {
    background-color: #0056b3;
}
#label-container {
    margin-top: 20px;
    width: 100%;
    max-width: 300px;
    text-align: center;
}
#label-container div {
    background-color: white;
    padding: 10px;
    margin: 5px 0;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0,0,0,0.1);
}
```

- மேல் உள்ளவாறு css உருவாக்கிய பின் உங்களுக்கு web site பின்வருமாறு தோன்ற வேண்டும் .

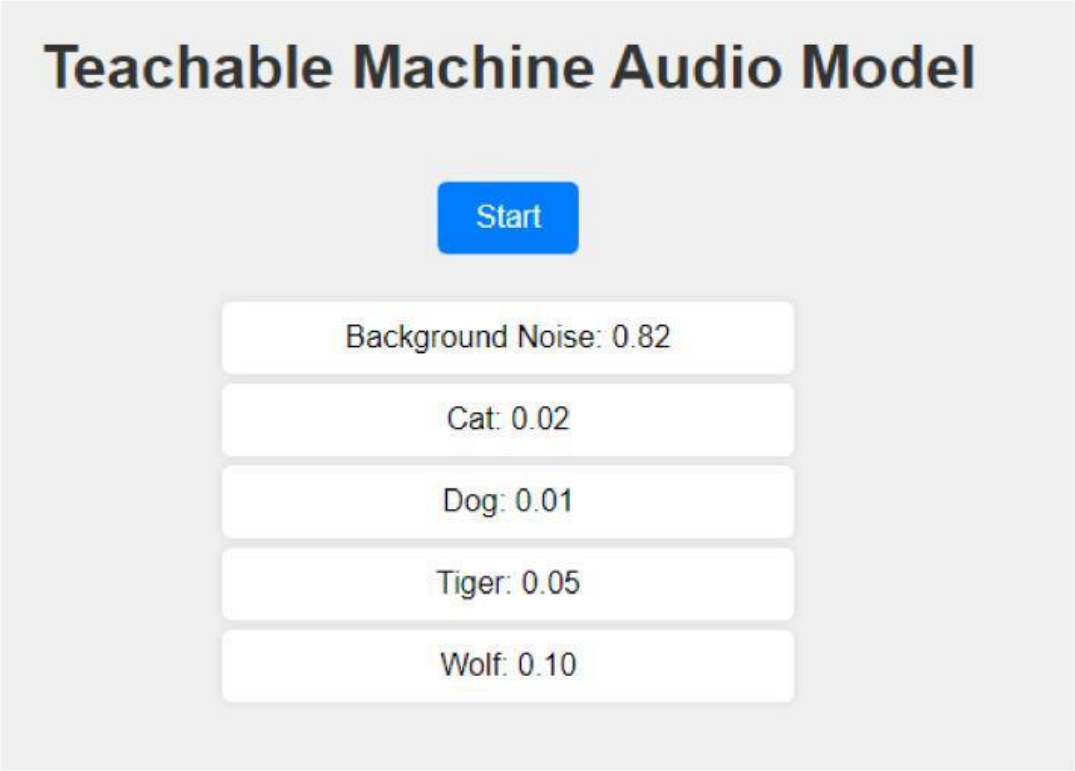


Start button இற்குள் click பண்ணும் போது microphone இற்கு access browser மூலம் பெரும்பாலும் request பண்ணுவது நடைபெறும். அதற்கு allow பெற்றுக் கொடுக்க .

LIVEWORKSHEETS



❖ அப்போது பின்வருமாறு prediction model run ஆகுமாறு window தோன்றும்.



❖ தற்போது உங்களுக்குப் பெற்றுத் தரப்பட்டுள்ள ஏனைய audio மூலம் ஒவ்வொரு மிருகங்களினதும் ஒலியை play பண்ணி predict பண்ணுவதை இதன் மூலம் நிறைவேற்றிக் கொள்ள முடியும் .

LIVEWORKSHEETS