**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Stryker Teachathon</title>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<nav id="navbar" class="">

<div class="nav-wrapper">

<div class="logo">

<!-- Logo Placeholder for Inlustration -->

<a href="index.html"><i class=""></i><img src="Logo.png" width="200" height="80"></a>

</div>

<ul id="menu">

<li><a href="index.html">Home</a></li>

<li><a href="demo.html">Demo</a></li>

<li><a href="about.html">About</a></li>

</ul>

<div>

</nav>

<div class="content">

<h2 class="title"><span style="color: #FAAB05;">Stryker</span> Techathon</h1>

<div class="idea">

<br>

<p>Hip or Knee replacement post surgery issue identification using Computer Vision</p>

</div>

</div>

</body>

**Demo.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Stryker Teachathon</title>

<link rel="stylesheet" href="style.css" />

</head>

<body onload="init()">

<nav id="navbar" class="">

<div class="nav-wrapper">

<div class="logo">

<!-- Logo Placeholder for Inlustration -->

<a href="#home"><i class=""></i><img src="Logo.png" width="200" height="80"></a>

</div>

<ul id="menu">

<li><a href="index.html">Home</a></li>

<li><a href="demo.html">Demo</a></li>

<li><a href="#about">About</a></li>

</ul>

<div>

</nav>

<div class="below-nav">

<div class="vision">

<div><canvas id="canvas"></canvas></div>

<div id="label-container"></div>

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose.min.js"></script>

<script src="script.js"></script>

</body>

**Script.js**

const URL = "https://teachablemachine.withgoogle.com/models/yeJj93Mq5/";

let model, webcam, ctx, labelContainer, maxPredictions;

async function init() {

const modelURL = URL + "model.json";

const metadataURL = URL + "metadata.json";

// load the model and metadata

model = await tmPose.load(modelURL, metadataURL);

maxPredictions = model.getTotalClasses();

// Convenience function to setup a webcam

const size = 600;

const flip = true; // whether to flip the webcam

webcam = new tmPose.Webcam(size, size, flip); // width, height, flip

await webcam.setup(); // request access to the webcam

await webcam.play();

window.requestAnimationFrame(loop);

// append/get elements to the DOM

const canvas = document.getElementById("canvas");

canvas.width = 500; canvas.height = 550;

ctx = canvas.getContext("2d");

labelContainer = document.getElementById("label-container");

for (let i = 0; i < maxPredictions; i++) { // and class labels

labelContainer.appendChild(document.createElement("div"));

}

}

async function loop(timestamp) {

webcam.update(); // update the webcam frame

await predict();

window.requestAnimationFrame(loop);

}

async function predict() {

// Prediction #1: run input through posenet

// estimatePose can take in an image, video, or canvas html element

const { pose, posenetOutput } = await model.estimatePose(webcam.canvas);

// Prediction 2: run input through teachable machine classification model

const prediction = await model.predict(posenetOutput);

for (let i = 0; i < maxPredictions; i++) {

const classPrediction = prediction[i].className;

const probability = prediction[i].probability.toFixed(2) \* 100; // Convert probability to percentage

// Update the label with class prediction and probability

labelContainer.childNodes[i].innerHTML = `${classPrediction}: ${probability}%`;

// Create a visual bar

const bar = document.createElement('div');

bar.style.width = `${probability}%`;

bar.style.height = '20px'; // Adjust the height of the bar

// Set the bar color based on class (assuming the first class should be red)

if (i === 0) {

bar.style.backgroundColor = 'green';

bar.style.borderRadius = '30px';

} else {

bar.style.backgroundColor = 'red'; // Color other classes as red

bar.style.borderRadius = '30px';

}

// Append the bar to the labelContainer

labelContainer.childNodes[i].appendChild(bar);

}

// finally draw the poses

drawPose(pose);

}

function drawPose(pose) {

if (webcam.canvas) {

ctx.drawImage(webcam.canvas, 0, 0);

// draw the keypoints and skeleton

if (pose) {

const minPartConfidence = 0.5;

tmPose.drawKeypoints(pose.keypoints, minPartConfidence, ctx);

tmPose.drawSkeleton(pose.keypoints, minPartConfidence, ctx);

}

}

}

**About.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Stryker Teachathon</title>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<nav id="navbar" class="">

<div class="nav-wrapper">

<div class="logo">

<!-- Logo Placeholder for Inlustration -->

<a href="index.html"><i class=""></i><img src="Logo.png" width="200" height="80"></a>

</div>

<ul id="menu">

<li><a href="index.html">Home</a></li>

<li><a href="demo.html">Demo</a></li>

<li><a href="about.html">About</a></li>

</ul>

<div>

</nav>

<div class="developers">

<br>

<br>

<h1 class="dev" style="color: #FAAB05;">Developers</h1>

<div class="photos">

<div>

<img src="Akhil-Majety.jpg" class="pic" width="200" height="200">

<h1 class="names">Akhil Majety</h1>

<p class="role">IT Technical Analyst</p>

<p>Commerical International - Emerging Markets</p>

</div>

<div>

<img src="pallavi.jpg" class="pic" width="200" height="200">

<h1 class="names">Pallavi C</h1>

<p class="role">IT Technical Analyst</p>

<p>Commerical International - Emerging Markets</p>

</div>

</div>

</div>

</body>

**Style.css**

\* {

border: 0;

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

background: rgb(250, 250, 250) url('Ortho-Bg.jpg');

font-family: Lato, Helvetica, Arial, sans-serif;

}

a {

color: inherit;

font-family: inherit;

font-size: inherit;

text-decoration: none;

}

#navbar {

color: white;

position: fixed;

top: 0;

height: 60px;

line-height: 60px;

width: 100vw;

z-index: 10;

}

.nav-wrapper {

margin: auto;

text-align: center;

width: 70%;

}

.logo {

float: left;

margin-left: -180px;

margin-top: 18px;

}

#navbar ul {

display: inline-block;

list-style: none;

margin-top: -2px;

text-align: right;

transition: transform 0.5s ease-out;

-webkit-transition: transform 0.5s ease-out;

margin-top: 30px;

}

#navbar li {

display: inline-block;

}

#navbar li a {

color: white;

display: block;

font-size:medium;

height: 50px;

letter-spacing: 1px;

margin: 0 20px;

padding: 0 4px;

position: relative;

text-decoration: none;

text-transform: uppercase;

transition: all 0.5s ease;

-webkit-transition: all 0.5s ease;

}

#navbar li a:hover {

/\* border-bottom: 1px solid rgb(28, 121, 184); \*/

color: rgb(28, 121, 184);

transition: all 1s ease;

-webkit-transition: all 1s ease;

}

/\* Animated Bottom Line \*/

#navbar li a:before, #navbar li a:after {

content: '';

position: absolute;

width: 0%;

height: 1px;

bottom: -1px;

background: rgb(13, 26, 38);

}

#navbar li a:before {

left: 0;

transition: 0.5s;

}

#navbar li a:after {

background: rgb(13, 26, 38);

right: 0;

/\* transition: width 0.8s cubic-bezier(0.22, 0.61, 0.36, 1); \*/

}

#navbar li a:hover:before {

background: rgb(13, 26, 38);

width: 100%;

transition: width 0.5s cubic-bezier((0.22, 0.61, 0.36, 1));

}

#navbar li a:hover:after {

background: transparent;

width: 100%;

/\* transition: 0s; \*/

}

.content{

margin-top: 260px;

text-align: left;

margin-left: 150px;

}

.title{

color: white;

font-size:50px;

font-weight: bold;

margin-left: 200px;

}

.idea{

color: white;

font-size:30px;

font-weight: normal;

width: 60%;

text-align: center;

}

.name{

font-weight: bold;

margin-bottom: 5px;

}

.below-nav{

margin-top: 80px;

background-color: white;

height: 550px;

margin: 50px 90px 50px 90px;

border-radius: 8px;

}

.vision{

display: grid;

grid-template-columns: 1fr 1fr;

grid-gap: 5px;

margin: 120px 40px 20px 100px;

}

#label-container{

margin-top: 200px;

font-size: larger;

}

.developers{

margin-top: 100px;

background-color: white;

height: 550px;

margin: 100px 90px 50px 90px;

border-radius: 8px;

text-align: center;

}

.photos{

display: grid;

grid-template-columns: 1fr 1fr;

grid-gap: 0px;

margin-top:100px;

text-align: center;

}

.pic{

border-radius: 50px;

}

.role{

font-weight: 600;

}

.names{

padding-bottom: 5px;

}