

DETECÇÃO DE OBJETOS EM TEMPO REAL



Qual é o nosso OBJETIVO neste MÓDULO?

Aprendemos a importar um vídeo no nosso canvas p5.js e realizar a detecção de objetos nesse vídeo. Além disso, criamos um app web a partir dessas atividades.

O que APRENDEMOS na aula de HOJE?

- Projetamos um app web de IA de vigilância por vídeo.

Quais CONCEITOS/BLOCOS DE CÓDIGO estudamos hoje?

- Adicionamos o código para definir uma imagem de fundo para a página web;
- Adicionamos o código para criar uma tag heading e fornecemos estilos a ela;
- Adicionamos o código para o botão “Iniciar”;
- Adicionamos o código para criar um elemento HTML que contém a quantidade de objetos detectados e fornecemos estilos a ele;
- Adicionamos o código para criar um elemento HTML que contém o status da detecção e fornecemos estilos a ele;
- Adicionamos o código para a nota de copyright.

Como FIZEMOS as atividades?

1. Primeiro, adicione background (plano de fundo) à página web. Para isso, precisamos adicionar um **atributo background** dentro da tag body.

```
<body background="background.jpg">  
<center>  
| /center |
```

- Escreva um **atributo background** dentro da tag body e, dentro desse **atributo**, mencione o nome da imagem **background.jpg**. Precisamos colocar o mesmo nome da imagem presente na pasta **AI-Video-Surveillance**, a qual você fez download para esta aula.

Resultado:



- Como pode ver, a imagem de fundo é repetida, pois é muito pequena para cobrir toda a tela.
- Queremos que uma única imagem cubra toda a tela, logo, utilizamos a

propriedade de estilo `background-size:cover;`. Esta propriedade especifica o tamanho da imagem de fundo, redimensionando-a de acordo com o que mencionamos em CSS.

- Em CSS, mencionamos **cover** para `background-size`. Portanto, `background-size:cover;`.

cover: Redimensiona a imagem de fundo para que ela cubra toda a página web. Se a imagem for menor que o tamanho da tela, ela será alongada; se a imagem for maior que a tela, ela será recortada nas bordas.

```
body
{
    background-size: cover;
}
```

Resultado:



2. Em seguida, adicione uma tag `h1` para conter o cabeçalho/título do website. Além disso, forneça algumas classes Bootstrap.

```
<body background="background.jpg">
<center>
| <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
```

```
</center>
```

- **btn**: é uma class Bootstrap que adiciona padding (preenchimento) e margin (margem) a essa div, para que tenha uma aparência melhor.
- **btn-warning**: é uma class Bootstrap que adiciona a cor laranja claro ao fundo e à borda da tag.
- **heading**: é nossa class e fornecemos estilos a ela em **style.css**.

Estilos em **style.css**.

```
.heading
{
  font-size: 23px;
  width: 100%;
  margin: 0px;
}
```

- Definimos **font-size** (tamanho da fonte) para 23px;
- Definimos a **width** (largura) para 100%, a fim da heading cobrir toda a extensão da tela;
- Fornecemos **margin** para 0px, a fim de nossa heading tocar todas as extremidades da tela, para que não haja espaços vazios.

Resultado:

IA de Vigilância por Vídeo

3. Aqui, adicionamos duas tags br.

```
<body background="background.jpg">
<center>
    <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
    <br><br>
</center>
```

- Adicionamos essas duas tags br para que haja uma quebra de linha entre a tag h1 e a próxima tag do botão “Iniciar”.
4. Adicione um botão “Iniciar”. Esse será utilizado para iniciar o vídeo e o processo de detecção de objetos. Além disso, forneça classes Bootstrap e uma “id” a essa tag button (botão).

```
<body background="background.jpg">
<center>
    <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
    <br><br>
    <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
</center>
```

- **onclick = “start()”**: esse código chama a função start() quando o botão for pressionado. start() é utilizada para iniciar o vídeo e o processo de detecção de objetos. Definiremos essa função na próxima aula.
- **btn**: é uma class Bootstrap que adiciona padding e margin a esta div, a fim de fornecer uma aparência melhor.
- **btn-success**: é uma class Bootstrap que adiciona a cor verde clara de fundo e borda a essa tag heading.

- **id="start":** use esta id para fornecer estilos em style.css.

Estilos em style.css

```
#start
{
    font-size: 18px;
}
```

- Aumentamos o tamanho da fonte.

Resultado:



5. Aqui, adicionamos mais duas tags **br**.

```
<body background="background.jpg">
<center>
    <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
    <br><br>
    <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
    <br><br>
</center>
```

- Adicionamos essas duas tags br para que haja uma quebra de linha entre o a tag do botão “Iniciar” e a próxima tag h3 que contém a quantidade de objetos.

6. Adicione uma tag h3. O objetivo desta tag h3 é conter a quantidade de objetos detectados pelo modelo **CocoSSD**. Além disso, forneça classes Bootstrap e uma “id” a essa tag h3.

```
<body background="background.jpg">
<center>
    <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
    <br><br>
    <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
    <br><br>
    <h3 id="numberOfObjects" class="btn btn-warning"></h3>
</center>
```

- **id="numberOfObjects"**: id é utilizada para consultar esta tag h3, a partir de JS, enquanto ela for atualizada com a quantidade de objetos detectados. Ademais, forneça estilos a essa **id** em **style.css**.
- **btn**: é uma class Bootstrap que adiciona padding e margin a esta div e melhora sua aparência.
- **btn-warning**: é uma class Bootstrap que adiciona a cor laranja claro de fundo e às bordas da tag heading.

Resultado:

- Ainda não fornecemos um texto à tag h3, portanto, ela apresenta um espaço vazio.

Em style.css:

- Queremos que o mesmo estilo de fonte fornecido à tag HTML, com a id

```
#start
{
    font-size: 18px;
}
```

“start”,  seja aplicado na tag h3, que acabamos de criar. Para isso, adicione a id “numberOfObjects” ao lado da id **start**, junto de um separador “vírgula”.

```
#status , #numberOfObjects
{
    font-size: 18px;
}
```

- Com isso, fornecemos à tag h3 o mesmo estilo de fonte da id “start”.

7. Aqui, adicionamos mais duas tags **br**.

```
<body background="background.jpg">
<center>
    <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
    <br><br>
    <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
    <br><br>
    <h3 id="numberOfObjects" class="btn btn-warning"></h3>
    <br><br>
</center>
```

- Adicionamos essas duas tags br para que haja uma quebra de linha entre a tag h3, que utilizamos para conter a quantidade de objetos, e a próxima tag h3, que utilizamos para conter o status do modelo.

8. Em seguida, adicione uma tag h3 para conter o status da detecção dos objetos. Além disso, dê classes Bootstrap e uma id a essa tag.

```
<body background="background.jpg">
<center>
  <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
  <br><br>
  <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
  <br><br>
  <h3 id="numberOfObjects" class="btn btn-warning"></h3>
  <br><br>
  <h3 id="status" class="btn btn-danger"></h3>
</center>
```

- **id="status"**: esta id é utilizada para consultar a essa tag h3, a partir de JS, enquanto ela for atualizada com o status da detecção dos objetos. Além disso, forneça estilos a essa **id** em **style.css**.
- **btn**: é uma class Bootstrap que adiciona padding e margin a esta div e a deixa com uma aparência melhor.
- **btn-danger**: é uma class Bootstrap que adiciona a cor vermelha ao fundo e às bordas desta tag heading.

Resultado:

- Ainda não fornecemos um texto a esta tag h3, portanto, ela apresenta um espaço vazio.

Em style.css.

- Queremos que o mesmo estilo fornecido à tag HTML, com a id “start”

```
#status
{
  font-size: 18px;
}
```

, seja aplicado a esta tag h3 que acabamos de criar. Portanto, adicione a id “status” ao lado de **numberOfObjects** com um separador “vírgula”.

```
#start , #numberOfObjects
{
  font-size: 18px;
}
```

- Com isso, a tag h3, que acabamos de criar, terá o mesmo estilo de fonte das outras.

9. Aqui, adicionamos uma div para conter o rodapé de copyright do website.

```
<body background="background.jpg">
<center>
  <h1 class="btn btn-warning heading">IA de Vigilância por Vídeo</h1>
  <br><br>
  <button onclick="start()" class="btn btn-success" id="start">Iniciar</button>
  <br><br>
  <h3 id="numberOfObjects" class="btn btn-warning"></h3>
  <br><br>
  <h3 id="status" class="btn btn-danger"></h3>
</center>

<div class="copyrightFooter">
  Desenvolvido por -- Time BYJU'S
</div>
```

- Esse rodapé é um sticky footer (rodapé fixo). Ou seja, ele sempre estará posicionado em um ponto específico na base da página.

Você deve escrever seu nome.

10. Defina a div com nossa class.

```
<div class="copyrightFooter">
```

- **copyrightFooter**: é o nome da class da **div**. Dê estilos a esta **div** com a class.

11. Adicione estilos a esta div.

Para compreender uma sticky footer, execute este [código](#), e altere os valores a seguir para o rodapé.

```
<style>
  .footer {
    position: fixed;
    bottom: 0;
    width: 100%;
    background-color: red;
    color: white;
    text-align: center;
  }
</style>
</head>
<body>

<h2>Fixed/Sticky Footer Example</h2>
  ~~~~ m1 ~~~~ f1 ~~~~ t1 ~~~~ b1 ~~~~ ~
```

```
.copyrightFooter {
  position: fixed;
  color: #f5f5f5;
  bottom: 0;
  width: 100%;
  background-color: #6c757d;
  line-height: 2;
  text-align: center;
}
```

- Definimos a posição para fixed (fixo), pois queremos que essa div esteja fixa em uma posição do website;

`#6c757d;`

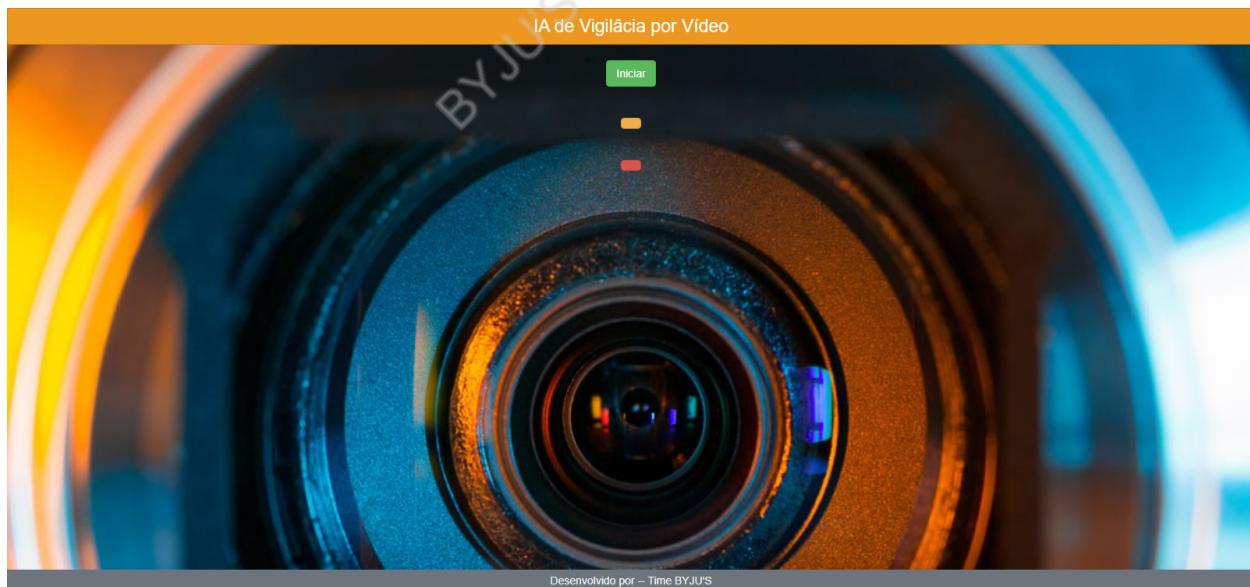
- Em seguida, definimos a cor do texto para `#6c757d;`, que é uma tonalidade cinza. Você pode utilizar qualquer cor;
- Definimos bottom (base) para 0, pois queremos que a div fique na parte inferior da página;
- Definimos a width para 100%, a fim de que ela cubra toda a width da tela horizontalmente;

`#f5f5f5;`

- Definimos a cor de fundo para `#f5f5f5;`, que é uma tonalidade clara de rosa. Você pode utilizar qualquer cor.
- Definimos line-height (limite de altura), que aumenta, um pouco, a height da div.
- Por fim, definimos o alinhamento do texto para “center”, com

`text-align: center;`

Resultado:



12. Na próxima aula, criaremos um elemento canvas p5.js, no entanto, forneceremos seu estilo agora. Aqui, finalizamos os códigos HTML e CSS. Para fornecer estilos ao

canvas p5.js, podemos escrever **canvas** e começar a adicionar as propriedades de estilo.

```
canvas
{
  border:2px solid black;
  border-radius: 10px;
  margin-top: 130px;
  box-shadow: 5px 5px 5px black;
}
```

- **border:2px solid white** resulta em uma borda de 2px com a cor branca.
- Adicionamos **border-radius: 10px;** para aplicar um efeito de bordas arredondadas.

Para entender box-shadow (sombra de caixa), execute este [código](#) e modifique os valores de box-shadow em style.css:

```
#example2 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px 18px yellow;
}
```

OBSERVAÇÃO:

quando executar o código, certifique-se de realizar os teste clicando no botão go live do visual studio. com isso, os arquivos serão executados no live server

do visual studio



Visto que utilizamos um arquivo de vídeo, e a biblioteca p5.js não nos permite

executar arquivos de imagem do sistema local, precisaremos executá-los em um server.

Qual é o PRÓXIMO passo?

Continuaremos o desenvolvimento desta aplicação web.

BYJU'S Future School