

HTML

Primeira linha -> dizer ao browser que tipo de documento é

<html> -> diz que é um html

<head> -> informação acerca do documento, tag do script dentro (código que se quer que se corra nesta página)

Variáveis: booleano, string, string, objeto

Console.log pode ser apagado linha 15

Console.log pode ser apagado linha 18

Console.log pode ser apagado linha 29

Linha 17 -> quando a página faz load executa o código até linha 39

Linha 19 -> websocket tem de estar na window (websocket deve vir do server)

Linha 20 -> app.route(camera) do server está-se a ligar ao html, procura uma ligação, (parte entre parêntesis é o caminho onde vai procurar a ligação)

Linha 21 -> está a usar o método onmessage, fica á espera de receber algo e depois executa o que está dentro da chaveta até linha 32

Linha 22 -> variável json_obj é a mensagem que vem da camera. Json é formato de dados, json.parse pega em string e transforma em objeto.

Linha 24 -> primeira parte procura o elemento que tem aquele id (ver cam lá em baixo), # significa id, atributo é src e basicamente vai por data:image na source lá me baixo na cam,

Ve a imagem e manda bué rápido para source, possibilitando ver o live feed

Linha 25 -> mandar string que é junção de json.aqc e json.final, enquanto não se carrega no botão ele fica a mandar o feed, caso carregue no botão recebe mensagem onde envia o prediction, está a mandar uma string em formato json

Linha 27 -> caso se carregue no botão ele mete na text box o valor da prediction

Linha 33 -> event handler que recebe erros em vez de mensagens como é o caso do event handler da linha 19 até linha 32, erro é a variável "e" que passa para função e é imprimido (na consola não no texto box)

Desde linha 19 até linha 36 é se o websocket estiver a funcionar, caso contrário diz que o websocket não é suportado

Function start click é quando o botão é clicado, manda info para @app.route(camera) do server, faz todo o processo do server.py e depois no html é reintroduzido na linha 21 mas desta vez acq_flag já não é false então faz o if(object.prediction) dando o resultado da prediction que depois é introduzido na text box.

<style> -> estética do html (cores, botões, caixas de texto, um elemento mais para cima ou mais para baixo, etc...)

-> id

. -> classe

"Ch" e "em" e "%" foram usados para ser mais responsivo do que usar pixéis, com pixéis as dimensões ficam as mesmas independentemente do ecrã, com os outros adaptam-se ao tamanho do ecrã

Float -> empurra o elemento para a direita ou para esquerda (em teoria também dá para cima e para baixo)

Position : relative -> foi necessário para por o texto debaixo do botão, relativo á div onde está.

Textarea não tem # nem . porque não é nem id nem classe e todos os elementos texto área vão ter este estilo

Retirar margin-left: 15%

<body> -> onde estão os elementos que vão ser visualizados na página

<div> -> tipo uma caixa

Linha 116 -> dimensões do ecrã/browser

Linha 118 -> onde vai buscar o logo fct