インターンシップ　プログラミング課題

顔画像解析システムを使ったアプリ開発体験

1. **???を正しいものに変えて、カメラの映像をWebページに出せるようにする(穴埋め)**

|  |
| --- |
| /\*\*  \* 課題(1)  \* 画像イメージをURLとして読み込む.  \*/  const imageLoader = imageElement => {  //課題(1) 下の???を正しいものに直す  const player = document.getElementById("**? ? ?**");  const reader = new FileReader();  reader.addEventListener("load", () => player.src = reader.result, false);  const file = imageElement.files[0];  if (file) {  reader.readAsDataURL(file);  }  }; |

1. **下のコードを追加して、撮影ボタンを押したら静止画を撮影して、映像の隣に表示できるようにする(写経)**

|  |
| --- |
| //課題(2) 撮影ボタンを押したら静止画を撮影して、映像の隣に表示する  //撮影ボタンを押したときに実行  const captureButton = document.getElementById('captureButton');  captureButton.addEventListener('click', function() {  const snapshot = document.getElementById('snapshot');  const context = snapshot.getContext('2d');  // スナップショットの表示  context.drawImage(player, 0, 0, snapshot.width, snapshot.height);  faceApi(snapshot.toDataURL('image/jpg'))  }); |

1. **下のコードの穴埋めをして、撮影した静止画を使って解析APIを呼び出せるようにする(穴埋め)**

|  |
| --- |
| //課題(3) 撮影した静止画を使って解析APIを呼び出せるようにする  const faceApi = function(sourceImage){  //スナップショット画像の取得  //FaceApi呼び出し  $.ajax({  url: uriBase + "?" + $.param(params),  beforeSend: function(xhrObj) {  xhrObj.setRequestHeader("Content-Type", "application/octet-stream");  xhrObj.setRequestHeader("Ocp-Apim-Subscription-Key", subscriptionKey);  },  type: "POST",  data: makeblob(sourceImage),  processData: false,  })  .done(function(data) {  // API呼び出し結果 画面表示  //➀textAreaという変数を作って、結果を表示したいHTML側の枠の要素（Id）を設定する  //ここに書く！  textArea.innerText = JSON.stringify(data, null, 2);  // 枠線表示  drawFaceline(data);  })  .fail(function(error) {  alert("error");  console.error(error);  });  } |

例）const A = document.getElementById('**B**');

HTML側（document）の**B**(Id)の要素を取得(getElmenetById())して変数A(const A)に設定している。

**4. 下のコードを穴埋めしながら書いてみて、静止画の顔のところに枠を出せるようにする(穴埋め)**

|  |
| --- |
| //課題４　顔に枠を出す処理部分を書き、解析結果表示の時に呼び出すようにする  //顔枠線処理//  const drawFaceline = function(data) {  data.forEach(function(element, index) {  //顔の位置座標を設定  const faceRe = element.faceRectangle;  ➀faceReに格納されている顔の位置情報を、それぞれleft,top,width,heightという変数を作ってそこに設定する。  ➁変数"canvas"を作り、顔枠を出したい静止画の要素を取得して設定する  const ctx = canvas.getContext('2d');  //枠線の色や太さの設定  ctx.lineWidth = 2;  //➂枠線の色を設定する（RGB方式）  ctx.strokeStyle = "rgb(ここに書く！)";  ctx.strokeRect(left, top, width, height);  // 番号を表示  　　　　　 //④枠線の上に表示する数字の色を設定する（RGB方式）  ctx.fillStyle = "rgb(ここに書く！)";  ctx.font = "bold 20px 'Arial'";  ctx.fillText(index + 1, left, top);  });  } |

1. **早く終わっちゃった人向け**

下のいくつかの課題から1つ選んで機能を追加してみよう

➀顔枠の上に年齢を表示してみる

➁幸せそうな顔かどうかを表示してみる

➂顔枠の上に一番強い表情が何かを表示してみる

➃顔の位置に別の画像を表示してみる

参考資料

https://azure.microsoft.com/ja-jp/services/cognitive-services/face/

https://docs.microsoft.com/ja-jp/azure/cognitive-services/face/quickstarts/javascript