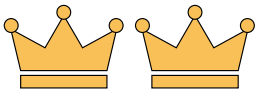
TECHNISCHE UNIVERSITÄT DRESDEN

Yoga Trainer



– Modell trainieren –

Professur für Didaktik der Informatik

In diesem Teil fangen wir an das Yoga-Projekt zu erstellen. Du findest eine Version hier. Du findest eine schon vorhandene Struktur vor. Schau dir diese erst einmal an. Die einzelnen Stellen sind mit Kommentare erklärt. Wenn du soweit bist, fangen wir mit dem ersten Schritt an.

VORBEREITUNG

Bevor wir mit dem eigentlichen Projekt anfangen, müssen wir noch einige Vorbereitungen treffen.

AUFGABE

Füge dein trainiertes Model von *Teachable Machine* in das Projekt ein. Ergänze dazu den Link zu deinem Model in der Variable `modelURL` ganz oben im Projekt.

Trage anschließend in das Array `figuren` alle Figuren ein, die du dem Model beigebracht hast.

HINWEIS

Das Modell von *Teachable Machine* wird über die *tmPose* **Bibliothek** bereitgestellt. Eine Bibliothek ist eine Sammlung an Funktionen und/oder Objekten. Mit dem Befehl

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

wird das Model geladen und ein **Objekt** `tmPose` bereitgestellt, welches wir der Variable `model` zuweisen.

SCHRITT 1: WEBCAM EINRICHTEN

Wir beginnen damit die *`init()`*-Funktion zu bearbeiten und das Model zu initialisieren. Als Erstes müssen wir natürlich die Webcam einrichten, damit wir ein Bild haben, dass das Modell auswerten kann. Unser Objekt `model` bringt dafür praktischerweise seine eigenen Methoden mit. Unter dem Punkt **Schritt 1** im Code initialisieren wir die Webcam.

AUFGABE

Zuerst legen wir die Größe der Webcam fest. Wir wählen dafür 400 Pixel.

```
const size = 400
```

Anschließend erstellen wir mit *tmPose* ein neues Webcam-Objekt. Die Parameter sind (Breite, Höhe, gespiegelt?). Da in *Teachable Machine* die Kamera quadratisch und gespiegelt war, erstellen wir auch das Webcam-Objekt entsprechend.

```
webcam = new tmPose.Webcam(size, size, true)
```

Anschließend müssen wir die Kamera vorbereiten, indem beispielsweise die Berechtigung zum Öffnen der Kamera beantragt wird. Anschließend starten wir die Kamera.

```
await webcam.setup()
await webcam.play()
```

DIE WEBCAM ANALYSIEREN

Nachdem die Kamera eingerichtet ist, kommen wir zum Kern der Applikation. Wir wollen unsere KI das Webcam-Bild analysieren lassen. Dies geschieht in der Funktion `predict()` unter **Schritt 2**.

AUFGABE

Damit die KI das Bild auswerten kann, müssen tatsächlich **zwei** Modelle hintereinander angewandt werden. Das erste erkennt das **Skelett** (die blauen Punkte und Striche). Dieses Modell haben wir nicht trainiert. Dieses ist bereits mithilfe von Millionen Daten vortrainiert worden.

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

Unser Model hat anhand des Skeletts gelernt und kann daran die Figuren erkennen. Wir übergeben dem Model deshalb nun das Skelett und speichern das Ergebnis unter `analyse` ab. Die Funktion `predictTopK()` gibt

das Ergebnis in Form eines **Arrays** zurück, in dem die wahrscheinlichste Pose an vorderster Stelle steht.

```
analyse = await model.predictTopK(posenetOutput)
```

FEEDBACK

Als nächstes müssen wir nur noch den Nutzer über das Ergebnis informieren und ihn damit motivieren seine Übungen durchzuhalten. Wir fügen den Code unter **Schritt 3** ein.

AUFGABE

Als nächstes geben wir ein Feedback über die analysierte Pose aus. Dazu geben wir in dem `<div> erkannteFigur` die analysierte Figur aus

```
erkannteFigur.innerHTML = analyse[0].className
```

Am Ende zeichnen wir das Skelett auf die Webcam. Dafür haben wir uns eine eigene Funktion definiert, die wir nur aufrufen und das Skelett übergeben.

```
poseZeichnen(pose)
```

Das Feedback soll sich natürlich mehr sein, als nur die erkannte Figur auszugeben. Dafür haben wir eine weitere Funktion `timerSteuern()` mit den wir den Ablauf des Timers regeln. Der Timer soll nur ablaufen, wenn die Figur gehalten wird. Das wollen wir jetzt einrichten. Dazu kommt noch kleine motivierende Nachricht, damit der Trainierende auch durchhält. Wir fügen den Code in der Funktion `timerSteuern()` unter **Schritt 4** ein.

AUFGABE

Wir überprüfen mit einer if-Verzweigung ob das erste Ergebnis aus `analyse` (also das wahrscheinlichste) die Figur ist, die gerade ausgewählt ist. Ist dem so, dann setzen wir die Variable `zeit` um eins nach unten. Je nach Ergebnis geben wir außerdem ein motivierende Äußerung aus. Hier kannst du auch kreativ sein!

```
if (figur == analyse[0].className){
      feedback.innerHTML = 'Super, du schaffst das!';
      zeit--;
}else{
      feedback.innerHTML = 'Versuche es noch ein
      bisschen!';
}
```

ZUSATZ

Vor dem Training könnte man mit einem Bot chatten und einen individuellen Trainingsplan entwickeln. Wie man einen Chatbot erstellt, erfährst du in der entsprechenden Station.

Wenn du möchtest kannst du deinen Yoga-Trainer hier oder zuhause noch weiter entwickeln.