

## Fruit to Emoji

Sebastian Stigler [sebastian.stigler@hs-aalen.de](mailto:sebastian.stigler@hs-aalen.de)  
Hochschule Aalen - Technik und Wirtschaft

---

## Was wird in diesem Aufgabenblatt alles gemacht

- Bibliothek für den **Farbsensor** wird installiert.
- Trainingsdaten aufgezeichnen.
- Das Machine Learning Modell wird trainiert.
- Die Machine Learning Applikation wird mit dem trainierten Modell implementiert.
- Die Applikation wird so angepasst, dass Emojis für die erkannten Früchte gesendet werden.
- Experimentieren.

## 1 Projekte laden und Bibliothek installieren

Starten Sie VSCode und fügen Sie die zwei Projekte (`object_color_capture` und `object_color_classify`) aus dem `platformio` Verzeichnis zum aktuellen Arbeitsbereich hinzu.

Anschließend installieren Sie die Bibliothek `Arduino_APDS9960` für den Farbsensor in die Projekte `object_color_capture` und `object_color_classify`.

## 2 Trainingsdaten aufgezeichnen

Ladend Sie das übersetzte Projekt Projekte `object_color_capture` auf den Mikrocontroller. Das Aufzeichnen der Trainingsdaten wird dann im Terminal in VSCode durchgeführt.

Zeichnen Sie nun die Daten für die verschiedenen Früchte auf. Für jede Frucht führen Sie bitte diese Schritte aus:

1. Drücken Sie die **Reset Taste** auf dem `Arduino Nano 33 BLE Sense`.
2. Rufen Sie `pio device monitor --quiet > frucht_name.csv` im Terminal von VSCode auf.
3. Führen Sie den Mikrocontroller nahe an die Frucht heran (mit der großen Öffnung im Gehäuse hin zur Frucht) und entfernen Sie ihn wieder. Wiederholen Sie das Ganze aus verschiedenen Richtungen rund um die Frucht herum ca. 50-70 mal.
4. Brechen Sie die Aufzeichnung im Terminal mit `Strg+c` ab.

## 3 Das Machine Learning Modell trainieren

Öffnen Sie die Colab (<https://colab.research.google.com>) und melden Sie sich mit Ihrem Googleaccount an.

Klicken Sie dort auf `Datei` → `Notebook hochladen` und laden Sie das Jupyter Notebook `colab/FruitToEmoji.ipynb` hoch. Folgen Sie nun den Anweisungen im Notebooks.

Das Notebook erzeugt die `model.h` Datei, die Sie in der nächsten Teilaufgabe benutzen werden.

## 4 Implementieren der Applikation mit dem trainierten Modell

Kopieren Sie die `model.h` Datei in das `src` Verzeichnis unter `platformio/object_color_classify`. Passen Sie das `CLASSES` Array derart an, dass Sie Ihren aufgezeichneten Früchten entspricht. Achten Sie darauf, dass die Einträge lexikographisch sortiert sind. Übersetzen Sie dann das Projekt und laden Sie auf den Mikrocontroller.

Starten Sie nun in VSCode den seriellen Monitor (Steckersymbol in der Fußleiste). Wenn Sie jetzt mit dem Mikrocontroller eine Frucht detektieren, dann sollten Sie im seriellen Monitor sehen, dass der Mikrocontroller diese Frucht erkennen kann.

## 5 Sende Emojis statt Text

Gehen Sie nun im Browser auf die Seite [https://image.informatik.htw-aalen.de/~stigler/jufo\\_display/](https://image.informatik.htw-aalen.de/~stigler/jufo_display/) und klicken Sie einmal in die Mitte der Seite.

Passen Sie nun das Projekt `object_color_classify` derart an, dass Sie **statt** der seriellen Ausgabe die Elemente aus dem Quellcode `Emoji_Keyboard` (siehe [02\\_GestureToEmoji](#)) benutzen, um die Ergebnisse der Klassifikation als Emojis auszugeben.

Sie benötigen dafür die Header `PluggableUSBHID.h` und `USBKeyboard.h` die bereits im Framework für den Mikrocontroller enthalten sind. Ferner benötigen Sie die `printString` Funktion sowie die Strings für die Emojis.

Die Strings sind dabei wie folgt aufgebaut:

- Er startet mit einem Großen **U**.
- Es folgen genau sechs hexadezimale Zahlen (Groß-/Kleinschreibung egal), die dem Code des Emojis entsprechen. Wenn der Code weniger als sechs Stellen hat, dann fügen Sie führende Nullen hinzu.

### Beispiel

Gegeben ist folgender Code für das Emoji:

Code	Emoji
U+1F600	😊

Dann verwenden Sie für den Keyboardstring den Wert `"U01F600"`.

Unter <https://www.unicode.org/emoji/charts/full-emoji-list.html> finden Sie eine Liste aller Emojis. Benutzen Sie die Browser Spalte um zu sehen, wie die Emojis dargestellt werden.

## 6 Experimentieren

Variieren Sie die Anzahl der Gegenstände die Erkannt werden sollen. Wieviele verschiedene Gegenstände kann das Modell noch sinnvoll unterscheiden?

Variieren Sie die Größe des Modells (mehr Layer, größe der Layer...). Wann macht der Mikrocontroller schlapp?