

# Emotion-Detection Roboflow Dataset

face emotion Classification Dataset (v1, 2022-12-28 8:39pm) by Hung  
5089 open source face-emotion images and annotations in multiple formats  
for training computer vision models. face emotion (v1, 2022-12-28 8:39pm),  
created by Hung

<https://universe.roboflow.com/hung-5yuey/face-emotion-8vfzj/dataset/1#>



## Klassifizierung auf Basis eines Roboflow-Dataset

Es soll ein Dataset von Roboflow genutzt und trainiert werden, um ein Modell für die Emotionserkennung zu erstellen, das anschließend in einem Anwendungsbeispiel verwendet wird.

### Technologien

- **TensorFlow:** Für die Erstellung und das Training des CNN-Modells.
- **Keras:** Hochwertige API für die Modelldefinition und das Training.
- **Roboflow:** Plattform zur Datenvorbereitung und -bereitstellung von vorab vorbereiteten Datasets.
- **H5:** Format für das Speichern und Weiterverwenden des Modells.

### Components

#### 1. Datenquelle: Roboflow

- Der Datensatz stammt von Roboflow und umfasst Bilder mit sieben verschiedenen Emotionen: anger, disgust, fear, happiness, neutrality, sadness,

surprise.

- Der Roboflow-Datensatz wurde in drei Ordner unterteilt: `train`, `valid`, `test`, wobei jeder Ordner in sieben Unterordner unterteilt ist (je einer für jede Emotion).

## 2. Datenvorbereitung

- Mit `ImageDataGenerator` aus Keras wurden die Trainings- und Validierungsdaten skaliert (Rescaling), um die Werte der Pixel im Bereich von 0 bis 1 zu normalisieren.
- Der `flow_from_directory` Befehl von Keras wurde verwendet, um die Bilder aus den Unterordnern direkt zu laden und in Tensoren zu konvertieren.

## 3. Modellarchitektur

- Das Modell basiert auf einem Convolutional Neural Network (CNN), bestehend aus:
  - **Convolutional Layers (Conv2D)**: Merkmale aus Bildern extrahieren.
  - **MaxPooling Layers (MaxPooling2D)**: Verbesserung der Modellgeschwindigkeit und Vermeidung von Overfitting.
  - **Dense Layers**: Um extrahierte Merkmale zu klassifizieren.
  - **Dropout Layer**: Verhindert Overfitting während des Trainings.
- Das Modell verwendet die `softmax` Aktivierungsfunktion in der letzten Schicht.

## 4. Speichern des Modells

- Nach dem Training wird das Modell im H5-Format gespeichert, um es später in anderen Anwendungen oder auf verschiedenen Plattformen zu verwenden.

## Ergebnisse

- Das Modell erreicht eine Accuracy von 75%, jedoch blieb die Validierungs-Accuracy nach ungefähr 40 Epochen, verbessert sich aber in der Val-Accuracy nach 7 Epochen kaum mehr und bleibt am Ende bei etwas über 62%, was darauf hinweist, dass das Modell noch nicht optimal funktioniert.
- Einige Emotionen wurden falsch klassifiziert, was die Genauigkeit des Modells beeinträchtigt.

## Ansätze

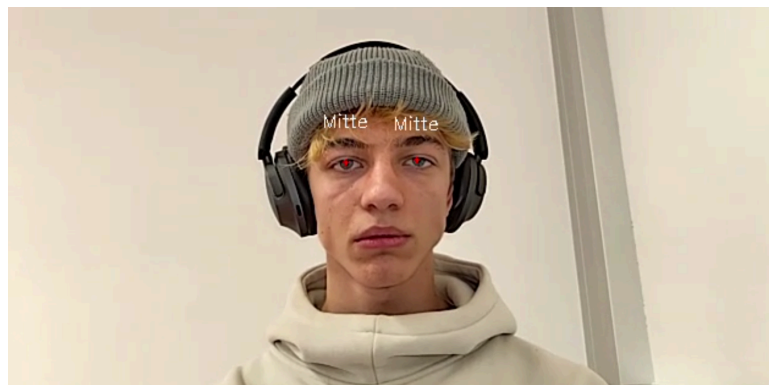
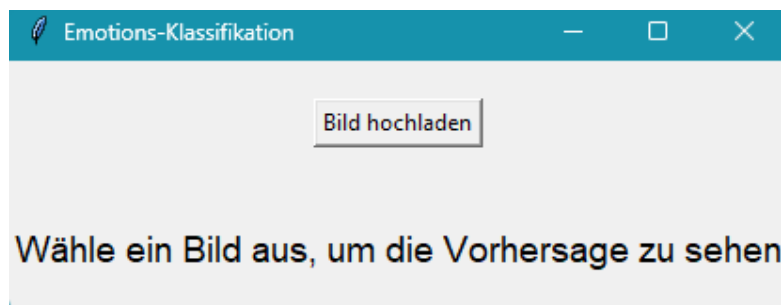
Zunächst habe ich versucht, das Modell mit dem YOLO-Ansatz (You Only Look Once) zu trainieren, um eine schnelle und effiziente Objekterkennung zu erreichen. Leider war die Leistung dieses Modells nicht zufriedenstellend. Daher habe ich mich entschieden, einen anderen Ansatz mit einem Convolutional Neural Network (CNN) zu verfolgen, da dieser besser für die Klassifikation von Emotionen aus Bildern geeignet ist.

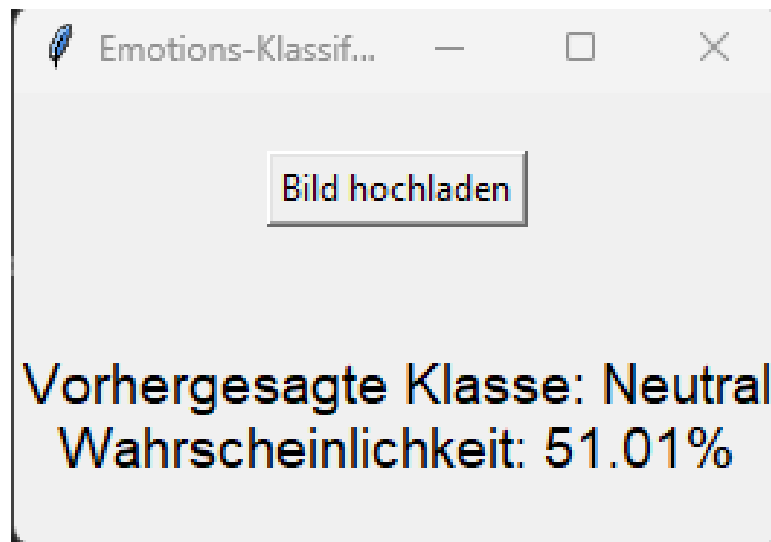
## Learnings

- **Datenvorbereitung ist entscheidend:** Eine ordnungsgemäße Vorbereitung und Normalisierung der Bilddaten ist eine der wichtigsten Voraussetzungen für das Training.
- **Hyperparameter-Anpassung:** Das Hinzufügen von Dropout und das Anpassen der Anzahl der Epochen hat das Modell schneller zu trainieren gemacht und zu einer besseren Leistung geführt.
- **Modellkomplexität und Overfitting:** Dieses Modell hat ohne Dropout zwar eine hohe Accuracy erreicht, war aber bei der Validierung weniger effektiv.
- **Probleme mit Python und Umgebungen:** Die Wahl der richtigen Python-Version und das Verwenden einer virtuellen Umgebung (venv) haben sich als entscheidend herausgestellt, da viele Probleme bei der Installation und Verwendung der Bibliotheken auftraten.

## Anwendung

Dazu habe ich einen Code geschrieben, der es ermöglicht, ein Bild hochzuladen und die Emotion darauf zu erkennen. Sobald ich ein Bild auswähle, verarbeitet das Programm es und zeigt die Vorhersage im Fenster an. Ich habe dafür Tkinter verwendet, um eine einfache Benutzeroberfläche zu erstellen. Ich habe es zuerst mit einer Webcam und Live Emotions Bestimmung versucht, was aber leider nicht funktioniert hat.





## Fazit

Es ist wichtig, den Datensatz korrekt vorzubereiten, geeignete Modelle zu wählen und die Hyperparameter anzupassen, um die Leistung zu steigern. Leider gab es Probleme bei der Modelloptimierung, was sich auf das Ergebnis auswirkte. Dennoch bietet der Ansatz eine solide Grundlage für emotionale Bildklassifikation und kann mit weiteren Anpassungen und Verbesserungen verbessert werden.

## Quellen

<https://universe.roboflow.com/hung-5yuey/face-emotion-8vfzj/dataset/1#>

<https://www.youtube.com/watch?v=wuZtUMEiKWY>

<https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-classification-on-custom-dataset.ipynb>