

# 

# Contents

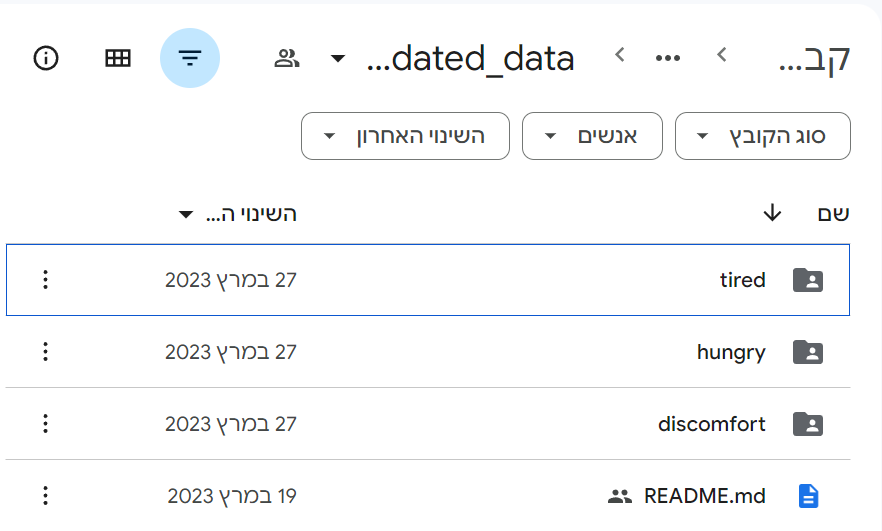
1. Initial Requirements
2. Model Training
3. App Installation

# Initial Requirements

* Internet connection - for downloading the model automatically by the app and running a production server.
* Trained model - h5 model file for predictions (explained in the next section).
* Python version 3.11 - recommended for running the app [Download from Microsoft store](https://apps.microsoft.com/store/detail/python-311/9NRWMJP3717K?hl=en-us&gl=us).
* ffmpeg installed - [Download from official website](https://ffmpeg.org/download.html).
* Add ffmpeg path to System Environment Variables in the Path section image
* MongoDB database - make sure to have Atlas account (for online database) or create localhost database (for local database) [MongoDB website](https://www.mongodb.com).

# Model Training

* Download a dataset or create your own - we used [donate-a-cry corpus dataset](https://github.com/gveres/donateacry-corpus). Important to note we used the **cleaned and updated version**.
* We combined burping and belly pain with discomfort and came up with 3 final labels - **Discomfort**, **Tiredness** and **Hunger**.
* The dataset is unbalanced and 2 out of the 3 labels have very few audio samples - this will require augmentations to expand those labels and dataset.
* Create a folder on you Google Drive and upload the dataset folders in this way:



* Create 2 additional folders (**Exported model** and **Spectrograms**).
* Make a copy of this Google Colab notebook [Crybaby model](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=ZWXv5pPWyQIg).
* Follow the step by step guide to create and train the model. ## Step-by-step guide for creating and training the model
* Set the WAV files directory to your **dataset** path and the results path to your **Spectrograms** path. Then run the code block to mount your Google Drive and save the paths. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=kkQlKiqy8_Yp&line=6&uniqifier=1).
* Run the code block to augment labels with very few samples using the audiomentations library, which applies transformations like Gaussian SNR, pitch shifting, time stretching, and normalization. The augmented files will be saved in the same directory with a suffix. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=Q3Jd4BP_GcFs&line=7&uniqifier=1).
* Set sample rate and duration parameters, perform data augmentation on specific subdirectories, load WAV files, adjust their duration, compute spectrograms for each window with overlap, and save the spectrograms as images using the ‘Magma’ color map. You can modify the augmented labels, which are preset to “Tiredness” and “Discomfort”. Run this code block only for the **first time!** [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=JvtDnJ0-Jlbs&line=3&uniqifier=1).
* Define a label-to-index mapping for the model’s classes, set paths for each label category, load images from the specified paths, define functions to load and display images, and initialize empty lists for storing image data and labels. Run the code block to define your labels and change the label paths if needed. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=rn9aYB375aZq&line=35&uniqifier=1).
* Run the next 3 code blocks to load files according to the label. [Code 1](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=d6oK-vjP54WS&line=3&uniqifier=1), [Code 2](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=T3oBv0Mh-yqU&line=2&uniqifier=1), [Code 3](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=0G3qNX6V-4OW&line=1&uniqifier=1).
* Perform train-test split on x and y data with stratification, normalize the training and testing data, and encode the training and testing labels using one-hot encoding. Run the code block to perform the split (you can modify the test set size according to your dataset size, we used 20%). [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=lPnataEA07kH&line=1&uniqifier=1).
* Run one of the model versions (Compact or Regular) to create the model. You can use any other model you wish. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=XcBpVhexEHGT&line=1&uniqifier=1).
* Train the model using the training data, evaluate its performance, plot the training and validation accuracy over epochs, save the accuracy plot as an image, evaluate the model’s loss and accuracy on the test data, generate a confusion matrix based on the model’s predictions, and visualize the confusion matrix as a heatmap plot. You can modify the hyperparameters to suit your needs and run the code block (remember to change the path to save the model plot). [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=ZWu43-5A4gxZ&line=7&uniqifier=1).
* Save the model in the desired path and with the desired name.
* **Optional:** To test the model on unseen data, follow these steps:
  1. Place your unseen data samples in a new folder on your Google Drive.
  2. Define the path for the unseen data samples and the results path in the next code block and run the code block. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=R3z9Kj5uENIn&line=5&uniqifier=1).
  3. The next code block will create spectrograms out of the unseen data. Run the code block (no augmentations needed for unseen data). [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=zC7VZUkpDwhE&line=1&uniqifier=1).
  4. The next code block loads and processes images from a specified directory, resizes them to a specific input size, converts them to numpy arrays, and uses a trained model to predict labels for the images. The predictions, filenames, and corresponding labels are written to a CSV file, and the results are printed for each image. Make sure to change the results path for the CSV file and run the code block. [Code](https://colab.research.google.com/drive/1B4__lcx4jVSa0GyM7LuUFc3F71G0ot5w?authuser=1#scrollTo=ZWXv5pPWyQIg&line=47&uniqifier=1).
  5. Open the CSV file to evaluate your model’s performance.

# App Installation

Clone repo:

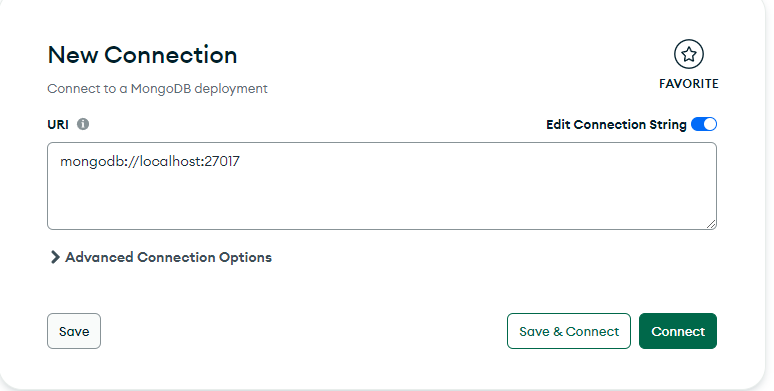
git clone https://github.com/maxko2/Crybaby-web-app

Install requirements.txt:

pip install requirements.txt

## Run app locally in development server

* Create a localhost connection on MongoDB - recommended to use MongoDB Compass [MongoDB Compass](https://www.mongodb.com/try/download/compass).



* Connect to localhost.
* On Crybaby.py and services.mongoDB.py use localhost connection string (‘mongodb://localhost:27017/’).

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

* Run Crybaby.py - using either **app.run(debug=True)** or **app.run(debug=True,host=“0.0.0.0”)**.
* Wait for model file to download.
* To start the app connect to **localhost:5000** or the **generated links(s)** - a database will be created with the name **Crybaby** and the collection **users**. If for some reason you are encountering a problem, create those manually.

## תמונה שמכילה טקסט, צילום מסך, גופן התיאור נוצר באופן אוטומטי

## Run app locally in production server

* Open port 5000 or any other port you wish to be using, on your router interface.
* Create a connection on MongoDB with **Atlas connection string** (or use localhost) - recommended to use MongoDB Compass [MongoDB Compass](https://www.mongodb.com/try/download/compass).

תמונה שמכילה טקסט, צילום מסך, תוכנה, מערכת הפעלה

התיאור נוצר באופן אוטומטי

* Connect to your database.
* On Crybaby.py and services.mongoDB.py use your Atlas connection string (or localhost string) - **follow local server instructions**.
* Run Crybaby.py - using **serve(app, host=‘0.0.0.0’, port=5000)** - Wait for model file to download.
* To start the app connect to **localhost:5000** or the **generated links(s)** - a database will be created with the name **Crybaby** and the collection **users**. If for some reason you are encountering a problem, create those manually.
* For other users to connect to your app use **external\_ip:port**:

**external\_ip**=[find your external ip](https://whatismyipaddress.com)

**port**=5000(default)

## Alternative:

Run server.bat.

## Optional

Use Ngrok to create free public domain and listen to your port [Ngrok download](https://ngrok.com/download).

ngrok.exe http 5000

תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי