Recognizing Facial Expressions Using a Convolutional Neural Network Model

Learning facial expressions from an image

Angry Disgust Fear Happy Neutral Sad Surprise

Team A:

- Chris Morgan

- Gregory Morales

- Naomi Shields

- Regina Negrycz

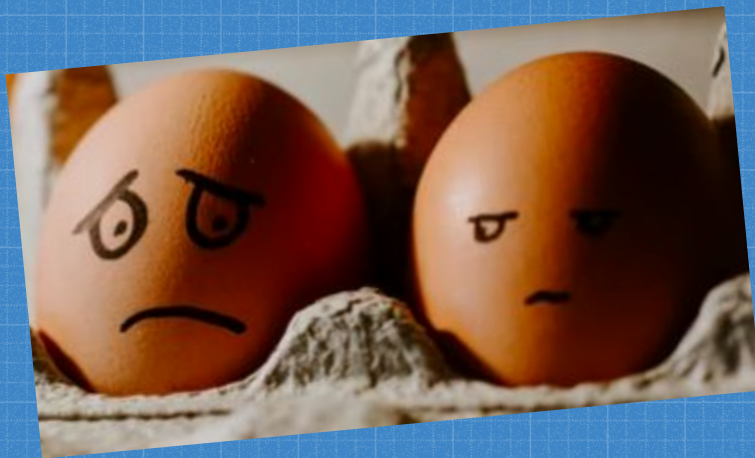# Dataset Chosen

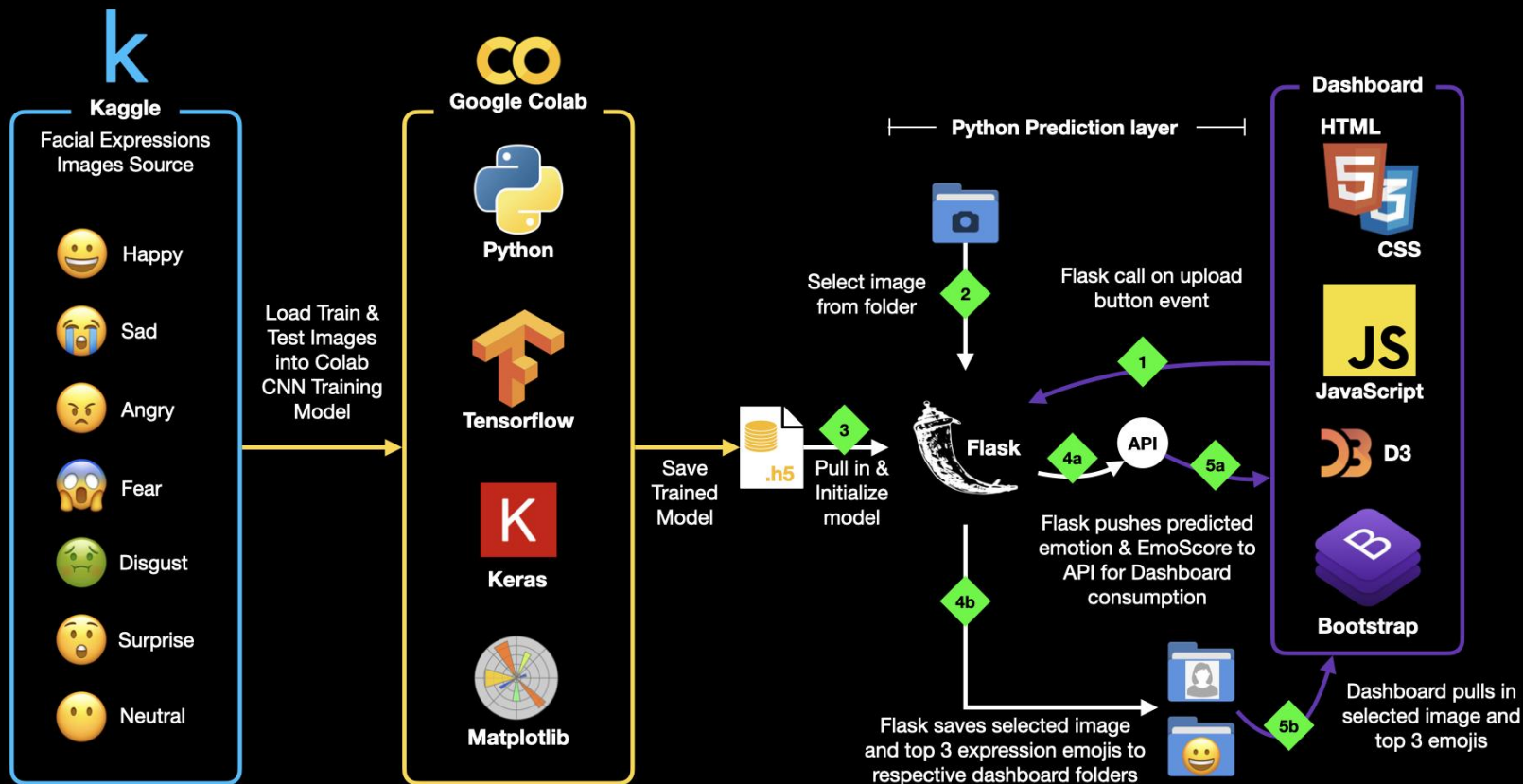Kaggle dataset FER-2013 consists of 48x48 pixel grayscale images of faces.

The faces are centered and occupy about the same amount of space per image.

Training set: 28,709 images
Test set: 3,589 images

Emoji the Possibilities: Convolutional Neural Network (CNN) Model

Kaggle

Facial Expressions Images Source

😀 Happy

😭 Sad

😠 Angry

😨 Fear

🤢 Disgust

😮 Surprise

😐 Neutral

Load Train & Test Images into Colab CNN Training Model

Google Colab

Python

Tensorflow

Keras

Matplotlib

Save Trained Model

.h5

3 Pull in & Initialize model

Python Prediction layer

Select image from folder

2

Flask

4a → API → **5a**

Flask call on upload button event

1

Flask pushes predicted emotion & EmoScore to API for Dashboard consumption

4b

Flask saves selected image and top 3 expression emojis to respective dashboard folders

5b

Dashboard pulls in selected image and top 3 emojis

Dashboard

HTML5

CSS

JS JavaScript

D3

Bootstrap

Technologies

Coogle Colab:
- ★ Python
- ★ Tensorflow
- ★ Keras
- ★ Matplotlib

**Python Prediction Layer:**
- ★ **.h5 trained model**
- ★ **Flask**

Dashboard:
- ★ HTML
- ★ CSS
- ★ JavaScript
- ★ d3
- ★ Bootstrap

Machine Learning

Tensorflow will be used to build a CNN model that will be used to classify expressions of uploaded black/white headshots. Google Colab was used to code the model and import all necessary python libraries to build initial saved model in a hdf (.h5) file format.
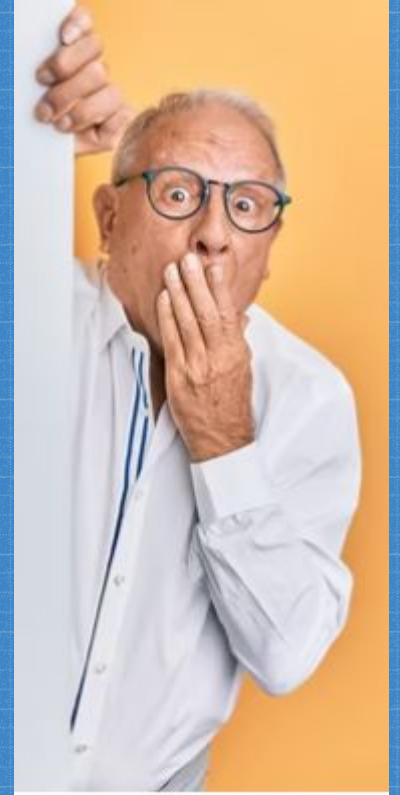
Benefits of CNN Models

Automatically detects the important features without human supervision.

High accuracy rate for image classification and recognition.

CNN is also proven effective with video, pattern and face recognition.

The algorithm is fast and simple.

Limitations of CNN Models

Missing pixels can have a detrimental effect on the verification process of the model.

If the CNN has several layers then the training process takes a lot of time if the computer doesn't consist of a good GPU.

Requires a large dataset to process and train the neural network.

Project Deliverables

Segment 2: Transform the database into a working model, refine the ML model, perform analysis and create visuals for story as well as begin dashboard.

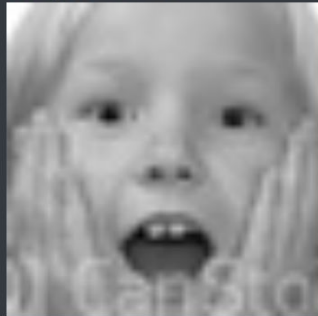Segment 3: Finalize database and ML models, create draft presentation and create the dashboard.

Segment 4: Final updates to README.md and final touches to presentation and dashboard. Practice presentation.

Functionality

The user will click on a button to upload a photo.

The upload will be the trigger to run the model

End Result:

The top 3 emotion predictions will be returned.