IRON HACK

Data Analytics

Is AI better at reading facial emotions than humans?

Beata Taudul

Feb, 2024

Table of content

Introduction

The Artificial Intelligence (AI) Emotion Recognition market is rapidly growing, with recent reports indicating its further increase in the coming years. AI systems are trained to analyze facial expressions, vocal intonations or sentiment analysis based on written text . By using machine learning algorithms, AI can identify patterns, improving accuracy in understanding of human emotions. This versatile technology finds applications in various sectors, including marketing, customer service, education, healthcare, virtual assistance, law enforcement, security, and entertainment.

Facial emotion recognition is a significant part of AI market. AI algorithms are trained to detect subtle changes in facial features and identify emotions like happiness, sadness, anger, and surprise. By analyzing the arrangement of facial muscles and comparing them to a vast database, AI can accurately predict a person's emotional state. But the question is, how good is it really?

The main goal for this project is to determine whether or not AI can be as good as human or better in facial emotion recognition. . In theory, AI is not (or should not be) influenced by subjective biases. However, what factors affect the model? How does the quality and quantity of data impact the results?

The first part of the project describes details of data gathering and analysis, emphasizing the role of emotions in daily life and the complexity of facial expressions. In the second part, a machine learning model will be trained using collected data, and the results will be compared with publicly available models, such as the DeepFace module.

The main points of the project are as follows:

- **Exploration of Topic:**
  Identify key factors related to the project's subject matter.
- **Data Collection:**
  Collect data from various sources, including web scraping, APIs, and flat files.
- **Data Cleaning and Exploratory Data Analysis (EDA)**:
  Perform data cleaning and conduct exploratory data analysis (EDA) using Python.
- **Database Creation:**
  Create a database to store and manage the cleaned data.
- **Entity Relationship Diagram (ERD):**
  Create EDR to show relationships between tables in database.
- **API Development:**
  Build an API to expose processed data and enable access for other users.
- **Machine Learning Model Training:**
  Train a machine learning model and compare its performance with available models.

Data and data sources

To make facial emotion recognition work, pictures of faces with the right labels for each emotion are needed. Research showed that common emotions recognized by AI include anger, disgust, fear, happiness, sadness, surprise, and a neutral face. These are the basic emotions that can be recognize by face expressions and are assumed to be universal for everyone.

To highlight the importance of emotions in our lives, I referred to a research paper that studied emotions in everyday life and the data it shared. I also gathered extra information about the Facial Action Coding System (FACS), which categorizes various facial movements and features linked to emotions. This matters for facial emotion recognition systems because it's a method to understand and mimic human-like emotions. Animations use this system to make emotions look realistic, and some algorithms might also use it to better understand and represent facial expressions.

The data sources for this project include:

Flat files:

- **Kaggle**

https://www.kaggle.com/datasets/sudarshanvaidya/random-images-for-face-emotion-recognition?select=anger

- **FER-2013**

https://www.kaggle.com/datasets/msambare/fer2013?select=train

- **Emotions in everyday life:**

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4689475/#:~:text=The%20most%20frequent%20emotion%20was,negative%20emotions%20simultaneously%20relatively%20frequently

API:

PixaBay: https://pixabay.com/api/

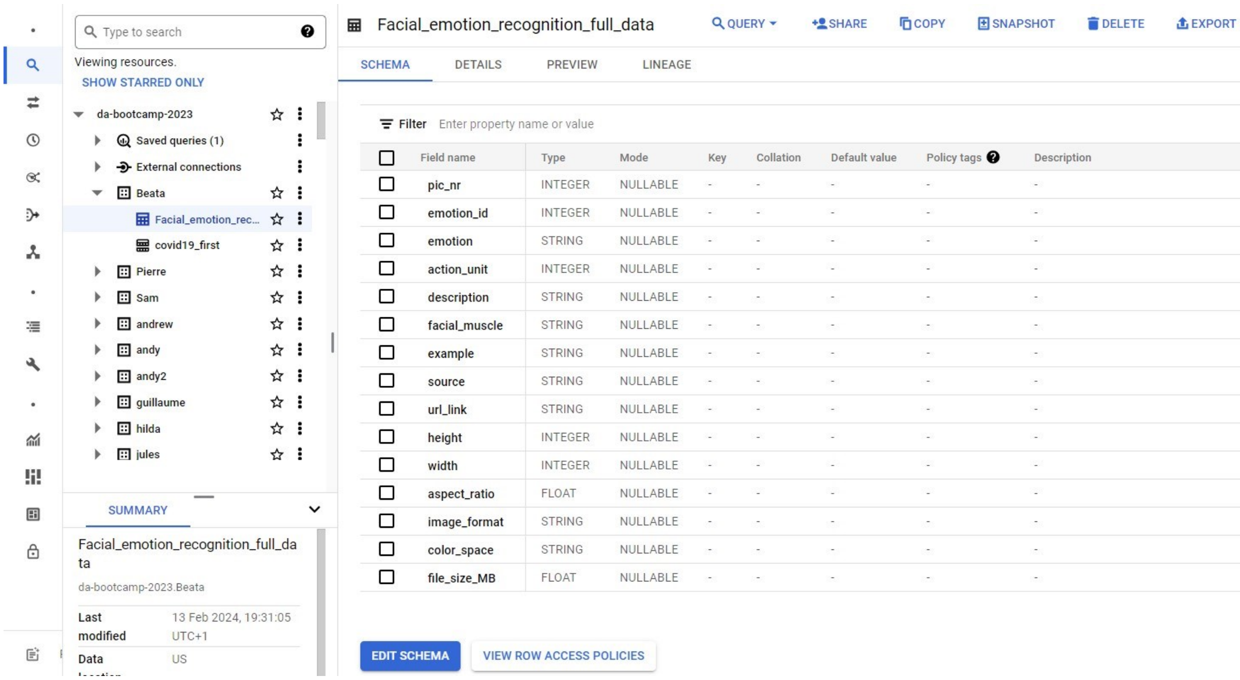
DuckDuckGo: https://duckduckgo10.p.rapidapi.com/search/images

Web Scrapping:

https://imotions.com/blog/learning/research-fundamentals/facial-action-coding-system/

Big Querry:

Unfortunately, there wasn't any available data directly associated with my project. Instead, I consolidated all the gathered data related to images and facial action units into a single comprehensive table.


Type to search
Viewing resources.
SHOW STARRED ONLY
da-bootcamp-2023
Saved queries (1)
External connections
Beata
Facial_emotion_rec...
covid19_first
Pierre
Sam
andrew
andy
andy2
guillaume
hilda
jules
SUMMARY
Facial_emotion_recognition_full_data
da-bootcamp-2023.Beata
Last modified 13 Feb 2024, 19:31:05 UTC+1
Data location US
Facial_emotion_recognition_full_data
QUERY
SHARE
COPY
SNAPSHOT
DELETE
EXPORT
SCHEMA
DETAILS
PREVIEW
LINEAGE
Filter Enter property name or value
Field name Type Mode Key Collation Default value Policy tags Description
pic_nr INTEGER NULLABLE - - - - -
emotion_id INTEGER NULLABLE - - - - -
emotion STRING NULLABLE - - - - -
action_unit INTEGER NULLABLE - - - - -
description STRING NULLABLE - - - - -
facial_muscle STRING NULLABLE - - - - -
example STRING NULLABLE - - - - -
source STRING NULLABLE - - - - -
url_link STRING NULLABLE - - - - -
height INTEGER NULLABLE - - - - -
width INTEGER NULLABLE - - - - -
aspect_ratio FLOAT NULLABLE - - - - -
image_format STRING NULLABLE - - - - -
color_space STRING NULLABLE - - - - -
file_size_MB FLOAT NULLABLE - - - - -
EDIT SCHEMA
VIEW ROW ACCESS POLICIES

Data cleaning

IMAGES from Kaggle and FER-2013

Kaggle (~43 MB) :

- over 5,300 + images with 7 emotion categories – anger, contempt, disgust, fear, happiness, neutrality, sadness and surprise,
- all images contain grayscale human face (or sketch),
- each image is **224 x 224 pixel grayscale** in PNG format,
- data for real human faces gathered from the internet and annotated manually. Images are sourced from the internet where they are freely available for download e.g. Google, Unsplash, Flickr etc.

FER-2013 (~43 MB) :

- over 28700 images with 7 emotion categories,
- all images of faces,
- each image is **48x48 pixel grayscale** in JPG format,
- faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image.

The initial number of pictures in each category for each data set can be found in table below:

Data set	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Kaggle	890	439	570	1406	524	746	775	5350
FER-2013	3995	436	4097	7215	4965	4830	3171	28709

- In both data set there is an imbalance among images of different emotions, 'Disgust' faces are a minority in both sets, while 'Happy' faces dominate the categories,
- To address this imbalance and prevent bias in the data, I randomly selected 200 pictures from each category in each dataset to balance the training data.
- I implemented a function that randomly picks images from each directory and saves them in a new one, utilizing the Python random module for this purpose.
- The number of selected images is flexible; I chose 200 pictures because I was uncertain about the capacity of the machine learning model on a local computer. Additionally, with the API, I couldn't retrieve as many pictures, and I didn't want them to be a significant minority in the data.
- Although the sizes of the images differ, it is not a concern at this stage. For the machine learning algorithm, the pictures will be normalized to have the same size.

IMAGES collected using APIs

Data retrieval with two step-process:

- a request was made to obtain images metadata and their URLs,
- the URLs were used to download and save images on disk (it was not possible to download all of the images retrieved in the first step)

Summary of number of images per category obtained through two APIs is presented in table below:

API		Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Pixabay	Initial	47	7	56	50	50	50	51	311
	After selection	26	5	7	36	33	17	23	147
	Face detection	**18**	**3**	**4**	**26**	**25**	**15**	**18**	**109**
DuckDuckGo	Initial	84	99	95	97	92	93	94	654
	After selection	73	71	70	89	59	71	88	521
	Face detection	62	63	55	79	58	51	80	**448**
	Total/category	80	66	59	105	83	66	98	

Challenges:

- with PixaBay API it was not possible to retrieve the same number of images for each category, in particular 'Disgust' was minority category in the beginning,
- obtained images are in color, with different resolutions, in JPG format,
- initial selection of images was required, over 130+ images for each Api was deleted,
  - part of the downloaded images were not suitable for further steps because face was not visible there or there were no humans at all in the images,
  - some of the images were not correctly assigned to emotion and had to be relabeled,
- then, the only the face had to be cut out from images for machine learning part:
  - I used pre-trained Haarcascade classifier from OpenCV for face detection,
  - algorithm was not always correctly detecting face in the image (even after tuning some of the parameters), but it allowed for automatic processing of hundreds of images,
  - the aspect ratio for processed images only with face visible was close to 1, although the height and width still varied between images (this will be normalized afterwords for machine learning part),
- about 200 images were dropped after selection and face detection in both groups,
- after cleaning total number of images per emotion category varies from 59 for fear to 105 for Happiness.

Creation of Data Frame with images:

- To construct a database with images, a function was created to transform images into a Data Frame,
- the function reads images from each emotion category and populates the Data Frame with metadata such as image source, expressed emotion, height, width, aspect ratio, color/grayscale, file format, and file size in MB,
- the last column of the Data Frame contains pixels of each image to allow for visualization,

Bolow is the example of created data frame. *Images_to_tables* is the function that does all the operations.

```
emotions = ['Anger', 'Disgust', 'Fear', 'Happiness', 'Neutral', 'Sadness', 'Surprise']
```

```
pixabay_path = '1_pixabay_images/1_Face_Extraction/'
```

```
pixabay = images_to_tables(pixabay_path, categories=emotions, source=1)
```

```
pixabay
```

	source_id	emotion_id	height	width	aspect_ratio	image_format	color_space	file_size_MB	pixels
0	1	0	302	302	1.000000	JPG	RGB	0.022323	[[[255, 255, 255], [255, 255, 255], [255, 255,...
1	1	0	1218	1280	1.050903	JPG	RGB	0.439991	[[[68, 68, 68], [73, 73, 73], [63, 63, 63], [8...
2	1	0	620	620	1.000000	JPG	RGB	0.136570	[[[231, 231, 231], [232, 232, 232], [233, 233,...
3	1	0	410	410	1.000000	JPG	RGB	0.041554	[[[36, 36, 36], [36, 36, 36], [36, 36, 36], [3...
4	1	0	320	320	1.000000	JPG	RGB	0.078756	[[[255, 255, 255], [126, 126, 126], [79, 79, 7...
...	...	...	...	...	...	...	...	...	...

CSV files with data from Web Scrapping

two tables were created:

- one containing single facial action units with their id, description, facial muscle involved, and example (gif) – total size (4, 25)
- second table with facial action units associated with each emotion, with size (2, 6) – table was then exploded to create a junction table between different emotions and single action unis,

CSV FILE 'Emotion_in_everyday_life_open.csv':

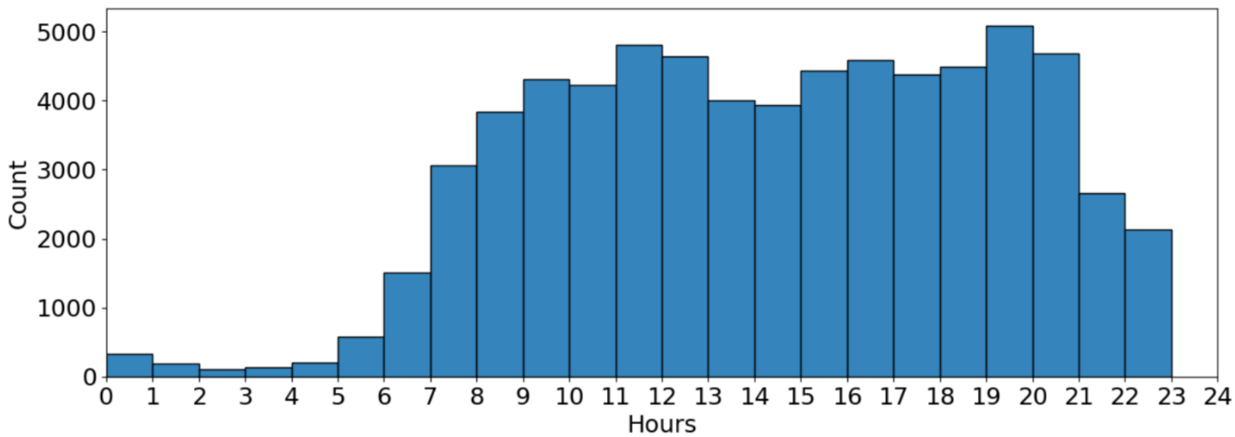
- initial size of the data set : (69544, 21)
- data type for columns were adjusted,
- columns with missing values values were dropped (1230 rows)
- columns are:

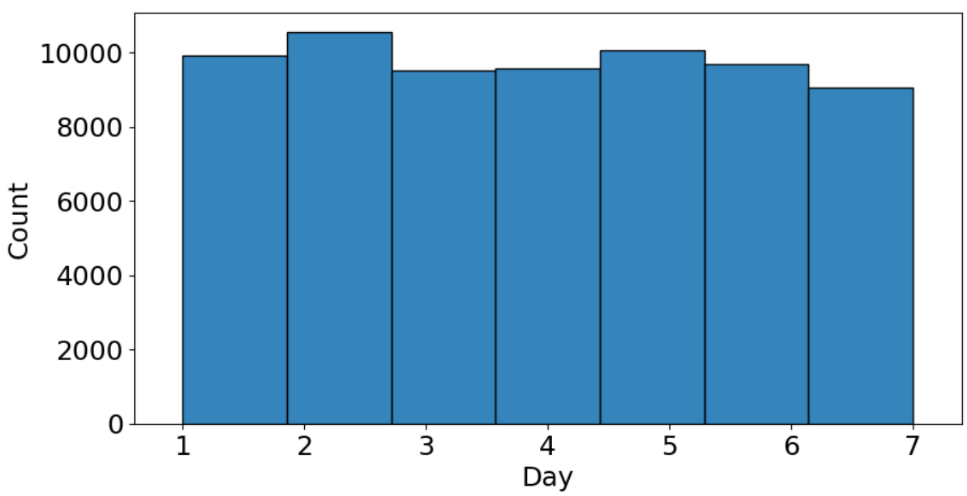
['id', 'Hours', 'Day', 'Pride', 'Love', 'Hope', 'Gratitude', 'Joy', 'Satisfaction', 'Awe', 'Amusement', 'Alertness', 'Anxiety', 'Disdain', 'Ofense', 'Guilt', 'Disgust', 'Fear', 'Embarassment', 'Sadness', 'Anger']

Exploratory data analysis

The data collected in 'Emotion_in_everyday_life_open.csv' provides statistics on the emotions people experienced throughout the day. A survey was randomly sent through the application to each participant, who then marked 0 or 1 based on whether they were experiencing a certain emotion or not.

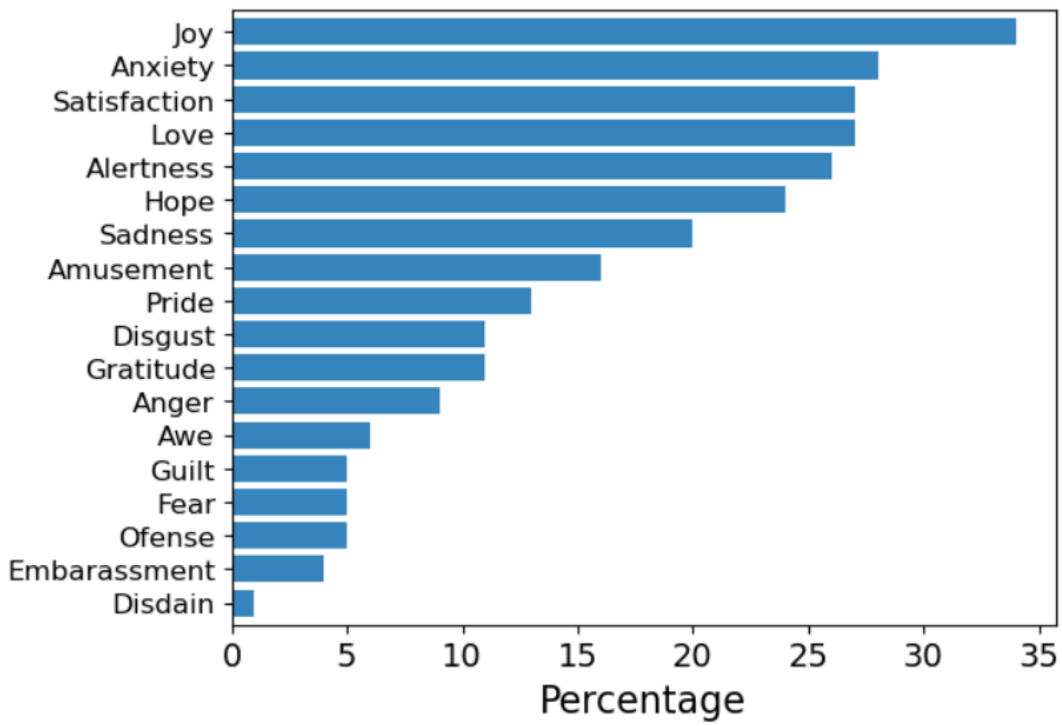
Number of questioners completed during different days of the week and times of the day





Number of questionnaires completed each day is well-balanced distribution across each day of the week, indicating consistent engagement throughout the entire week. In the case of different hours during a day, the peak activity occurs between 6 AM and 11 PM, while responses during night hours are minimal and will not be included in further analysis.

How do we feel during the day and over whole week?

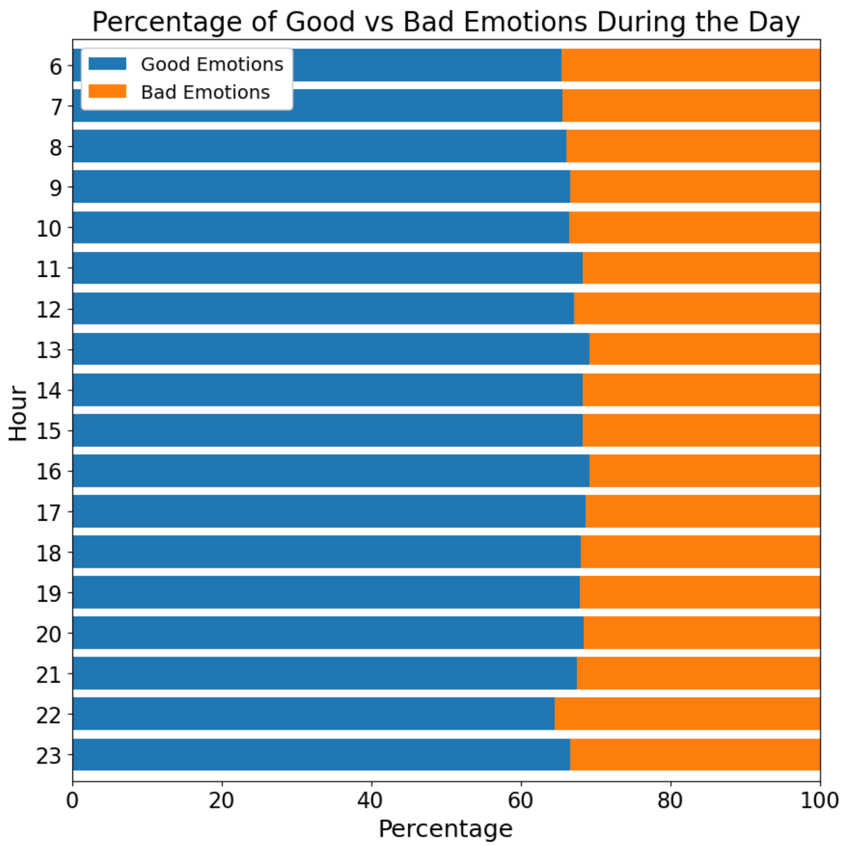


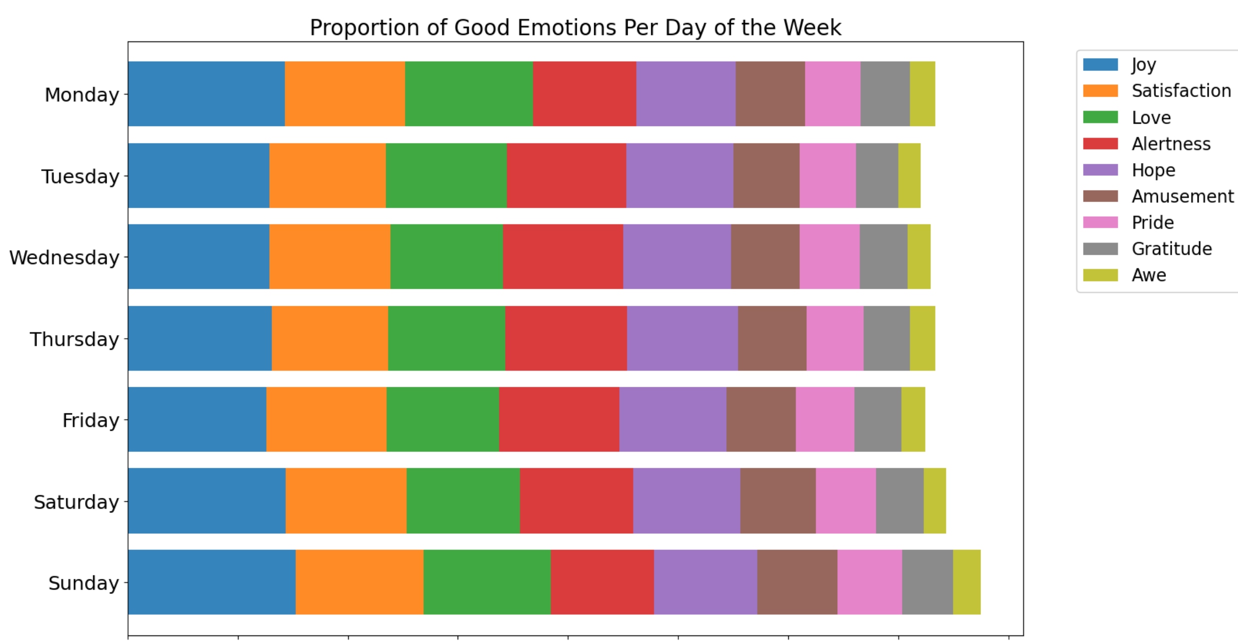
The data shows that Joy is the most common emotion, making up 34% of the occurrences. Anxiety follows closely at 28%, and Satisfaction and Love share the second spot with 27% each. Overall, the primary emotions are positive, except for a portion of Anxiety and Sadness.

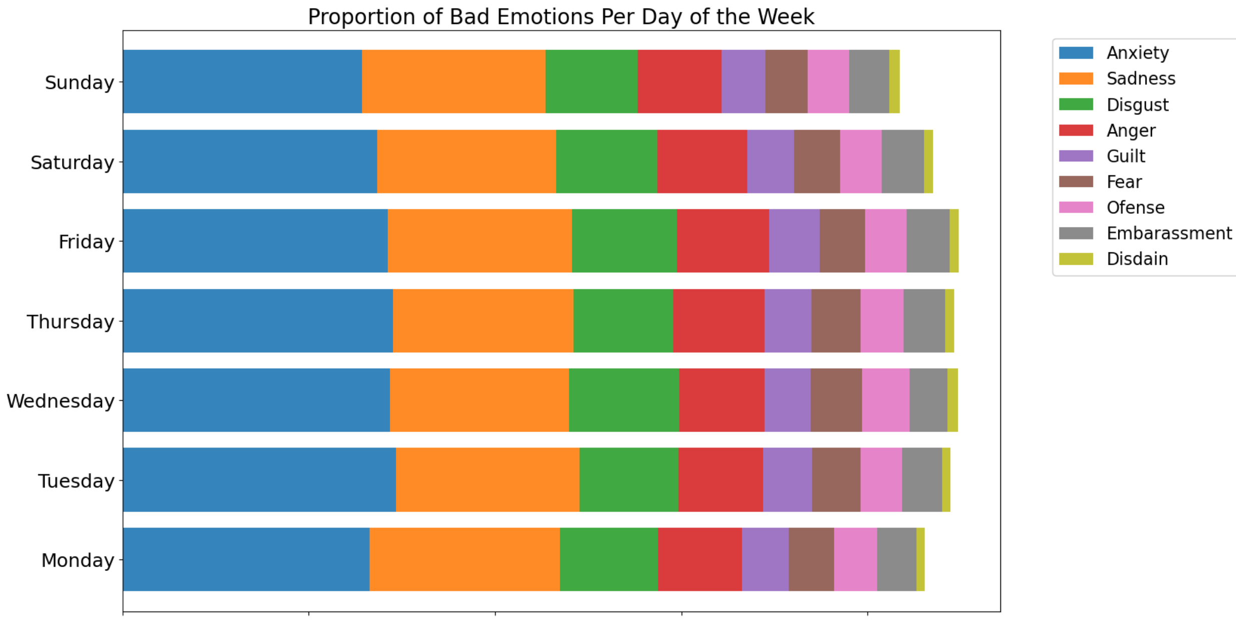
The data contained information regarding 18 different emotions. To facilitate analysis of emotional state during different times of the day, emotions were grouped as ‘bad’ and ‘good’ with each group containing 9 emotions.

Bad: ['Anxiety', 'Disdain', 'Ofense', 'Guilt', 'Disgust', 'Fear', 'Embarassment', 'Sadness', 'Anger’]

Good: ['Pride', 'Love', 'Hope', 'Gratitude', 'Joy', 'Satisfaction', 'Awe', 'Amusement', 'Alertness']


Percentage of Good vs Bad Emotions During the Day
Good Emotions
Bad Emotions
Hour
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
0
20
40
60
80
100
Percentage


Proportion of Good Emotions Per Day of the Week
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Joy
Satisfaction
Love
Alertness
Hope
Amusement
Pride
Gratitude
Awe



Data base type selection

In the data base world, there are two primary categories that define how data is stored and managed:

Relational Databases (SQL)

Relational databases structure data into tables comprising rows and columns. Each table corresponds to a specific entity, where rows represent instances and columns signify attributes. Relationships between tables are established through foreign keys. These databases leverage Structured Query Language (SQL) for operations like creation, reading, updating, and deleting data.

Non-relation Databases (NoSQL)

Contrasting relational databases, NoSQL databases do not adhere to a strict table-like schema. They handle various types of data, including unstructured, semi-structured, or structured data. NoSQL databases provide flexibility by not enforcing a rigid schema, allowing each entry to possess a unique structure. These databases utilize querying languages that typically differ from SQL.

Given the structured nature of my data, with images organized into tables and related information among each other, the choice of a Relational Database appears suitable. A relational Database would allow for reduction in data redundancy, and the ability to execute complex queries involving data from multiple tables using SQL.

Database creation

The data base was created using MySQL Workbench and called final_project.

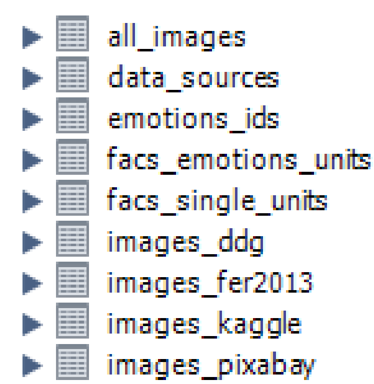
In Python, once the tables with image metadata were created, Python was connected to SQL, and the tables were pushed to the final_project database. Images from different sources were stored in separate data frames. This approach proved to be faster and more convenient than initially saving the data to CSV files and then loading it into MySQL Workbench.

A table containing data sources and their IDs was created using Python and sent to SQL Workbench. Similarly, a table containing a list of emotions and their IDs was prepared as a dataframe in Python and sent to SQL. The tables with sources and emotions were then linked to the tables with images by defining primary and foreign keys for all the tables.

Data scraped from a website, containing information about facial action units (FACS) and the FACS involved in each emotion, was imported from a CSV file. It was then connected to the emotion table using foreign keys. The relational diagram between all tables is presented later on.

In SQL, a table with all images was created to facilitate later use with Flask for data presentation. Otherwise, it would be necessary to create temporary tables with all images at each connection between Flask and SQL, which would be redundant.

List of tables:



Examples of SQL queries:

Count the number of images available for a chosen emotion, source, or with specific characteristics using the `all_images` table and a `WHERE` clause

```
SELECT COUNT(*) as total_rows
FROM all_images ai
JOIN emotions_ids ei ON ai.emotion_id = ei.id
WHERE ei.emotion = 'Anger';
```

Use subquery to select facial action units associated with particular emotion:

```
SELECT
 ei.emotion,
 action_unit,
 fs.description,
 facial_muscle,
 example
FROM (SELECT * FROM emotions_ids WHERE emotion = 'Anger') as ei
JOIN facs_emotions_units feu ON ei.id = feu.emotion
JOIN facs_single_units fs ON feu.action_units = fs.action_unit;
```

Count how many images are in given category based on their width and height and order them in descending order based on count in each group:

```
SELECT
height,
width,
COUNT(*) AS image_count
FROM all_images
GROUP BY height, width
ORDER BY image_count DESC;
```

Calculate total number of images in each emotion group:

```
SELECT
ei.emotion,
COUNT(*) as images_per_emotion
FROM all_images ai
JOIN emotions_ids ei ON ai.emotion_id = ei.id
GROUP BY ei.emotion
ORDER BY images_per_emotion
```

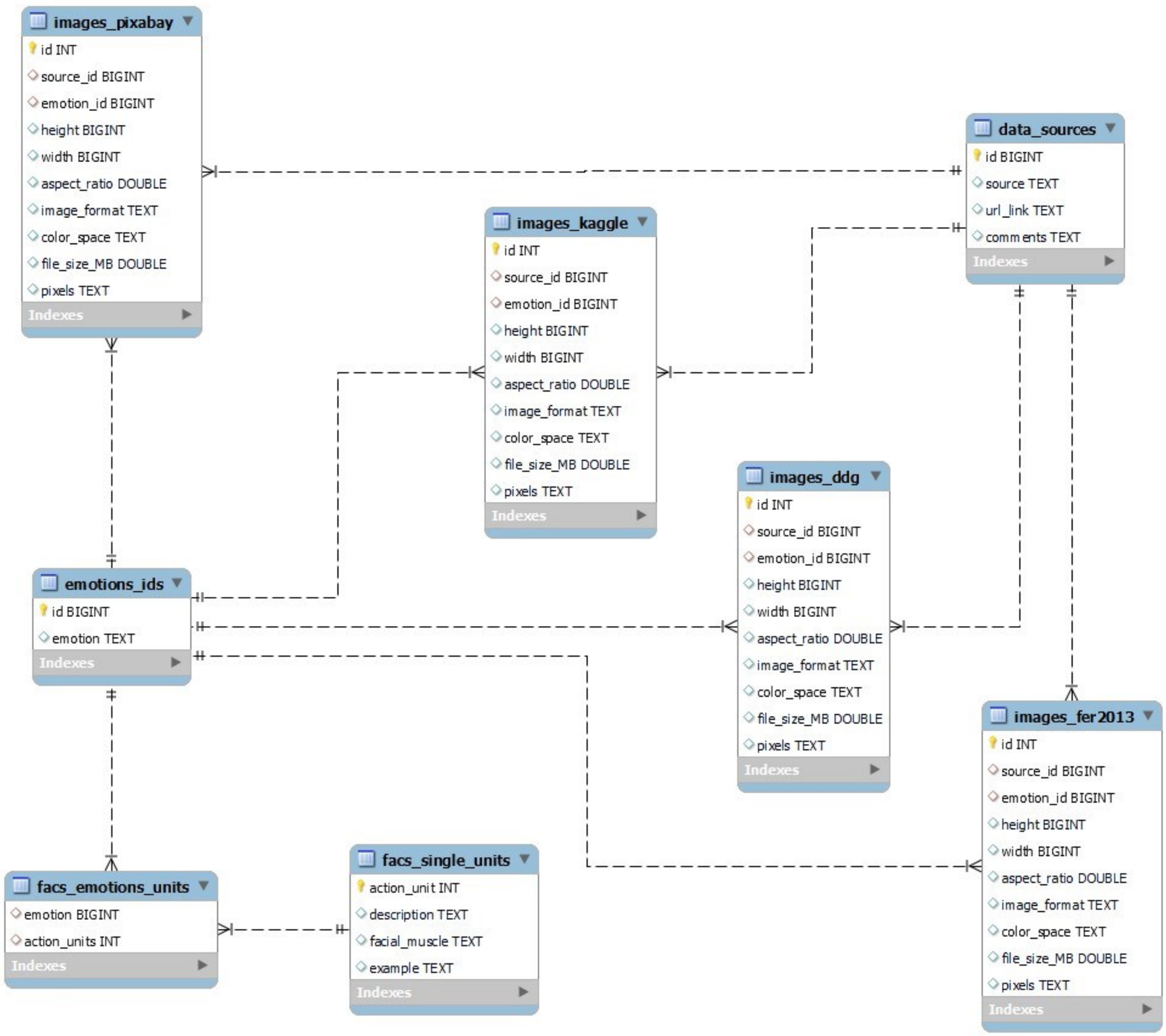
Find emotion than involves the highest number of facial action units:

```
SELECT
ei.emotion,
COUNT(*) AS nr_of_facial_units
FROM facs_emotions_units feu
JOIN facs_single_units fs ON feu.action_units = fs.action_unit
JOIN emotions_ids ei ON ei.id = feu.emotion
GROUP BY ei.emotion
ORDER BY nr_of_facial_units DESC
LIMIT 1;
```

Group images based on their aspect ration and classify them as suited or possibly problematic for machine learning:

```
SELECT
suited_for_ML,
COUNT(*) as total_number
FROM
(SELECT *,
   CASE
      WHEN aspect_ratio = 1 THEN 'should_be_fine'
      ELSE 'possible_distortions'
   END as suited_for_ML
FROM all_images) as ml
GROUP BY suited_for_ML;
```

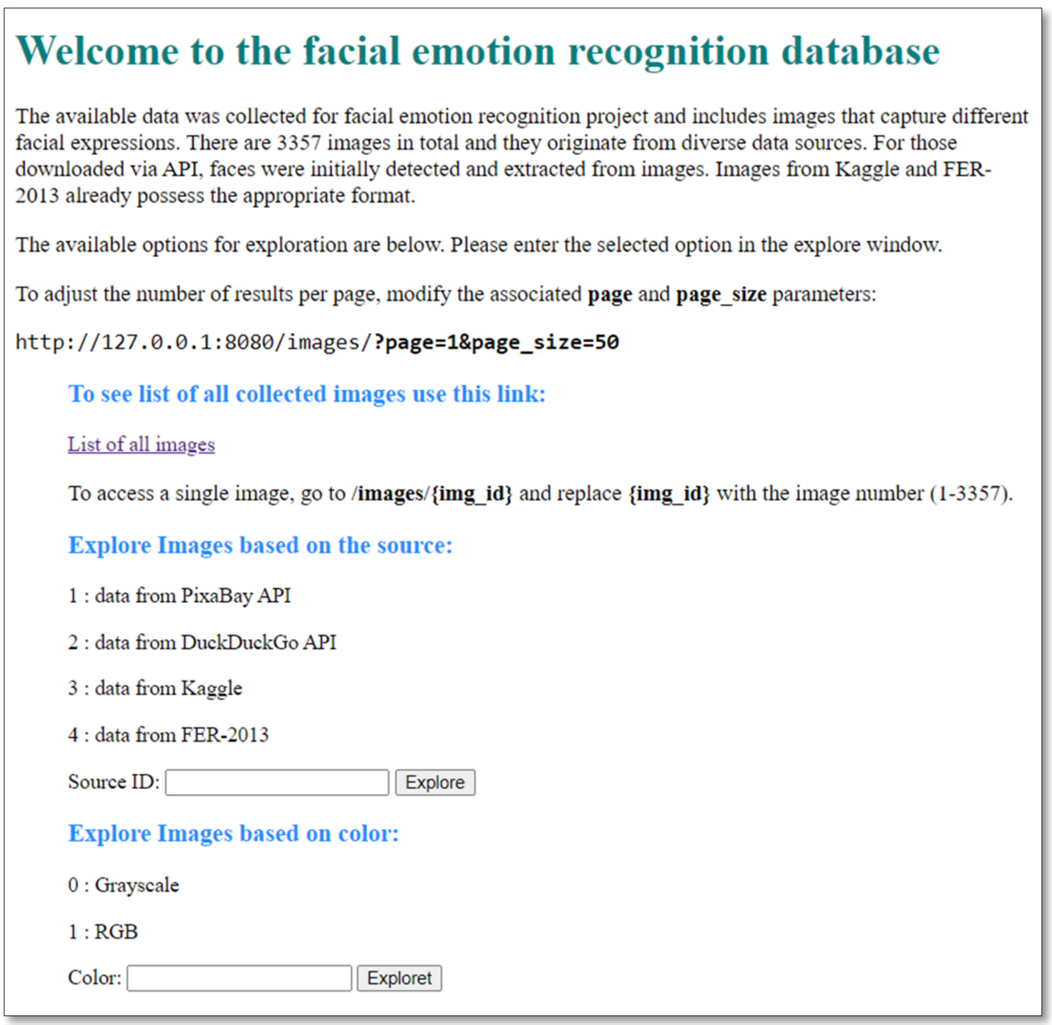
Entity Relationship Diagram (ERD



Exposing Data via API

The gathered dataset has been made accessible through an API developed using Flask, a lightweight and versatile Python web framework. Flask is chosen for its simplicity and flexibility, offering just the right tools for building web applications and APIs without unnecessary complexity.

To facilitate user interaction with the data, documentation for the API was prepared using an HTML template. This allows users to interactively specify most of the parameters for their searches. The access path has to be changed manually only if single image is requested or a user wants to adjust number of results per page. The home page, displayed after running the Flask API code, includes self-explanatory instructions on how to explore the dataset.



The API provides access to the stored data through following endpoints:

Access point	Description
/images/	Users can get access to data set with all available images. There are 3357 images collected. By default there are 50 results per page displayed, these values can be changed by user using access path.
/images/<int:image_id>	Gives possibility to show only one image at once by specifying its id. Possible values are 1-3357
/images/sources	Select images based on their source.
/images/emotions	Select images based on emotion shown. Isert corresponding number in the 'Emotion ID" field: 0-Anger, 1-Disgust, 2- Fear, 3-Happiness, 4-Sadness, 5-Surprise, 6-neutral
/images/aspect_ratio	Select images based on their aspect ratio. For most of them the aspect ratio is equal to 1, but there are also some images with smaller or higher values.
/images/colors	User can select between colorful and grayscale images
/images/emotions/faces	This access point allows to display facial action units associated with each emotion. It gives description of facial movement, names of muscles involved and an url link to a gif image that displays the facial expression.

How does it look like after search. Here is part of result after searching images based on emotion that is shown in the picture:


```
    {
      "aspect_ratio": 1.0,
      "color_space": "RGB",
      "emotion": "Surprise",
      "file_size_MB": 0.023431777954101562,
      "height": 196,
      "image_format": "JPG",
      "pixels": "[[[226 230 231]\n  [228 230 230]\n  [232 230 230]\n  ...\n  [ 47
64  85]\n  [149 159 176]\n  [234 243 255]]\n\n [[226 230 231]\n  [228 230 230]\n
[232 230 230]\n  ...\n  [ 47  65  88]\n  [ 77  92 111]\n  [210 223 239]]\n\n
[[226 231 230]\n  [228 230 230]\n  [230 230 230]\n  ...\n  [ 53  77 105]\n  [ 61
83 108]\n  [148 169 190]]\n\n ...\n\n [[ 50  64  87]\n  [ 46  63  89]\n  [ 24  48
78]\n  ...\n  [  2   4  15]\n  [  2   4  15]\n  [  2   4  15]]\n\n [[ 60  74
97]\n  [ 44  61  87]\n  [ 24  48  78]\n  ...\n  [  2   4  15]\n  [  2   4  15]\n
[  2   4  15]]\n\n [[ 59  73  96]\n  [ 32  49  75]\n  [ 39  62  94]\n  ...\n  [
2   4  15]\n  [  2   4  15]\n  [  2   4  15]]]",
      "source": "DuckDuckGo API",
      "url_link": "https://rapidapi.com/epctex-epctex-default/api/duckduckgo10",
      "width": 196
    }
  ],
  "last_page": "/images/emotions?emotion=5&page=10&page_size=50",
  "next_page": "/images/emotions?emotion=5&page=1&page_size=50",
  "previous_page": null
}
```

Summary

In conclusion, the AI Emotion Recognition market is experiencing rapid growth, acknowledging the increasing importance of emotions in everyday life and decision-making processes. However, the efficacy of these algorithms is significantly influenced by the quality and quantity of the gathered data and the methodology of their training.

Even in face recognition, challenges have emerged, indicating the necessity for continuous improvements. Human emotions, being inherently complex, pose a great challenge for accurate facial detection. Current discussions among scientists suggest the presence of 16, rather than 8, basic emotions recognizable through facial expressions. This emphasizes the existing limitations in AI, which is currently restricted to interpreting basic emotions.

As part of the machine learning aspect of this project, the goal is to build a customized model and assess its performance against existing ones.

The General Data Protection Regulation (GDPR)

GDPR, or the General Data Protection Regulation, is a European privacy regulation created to protect individuals' personal data. It sets clear rules for how organizations collect, process, and store personal information, ensuring transparency and accountability in data handling.

It's important to note that all the pictures gathered for this project were publicly available and did not contain any sensitive personal data. Similarly, the statistics collected for emotions in everyday life were sourced from an open-access article with publicly shared data. Any personal information was excluded from the shared data, ensuring compliance with privacy standards.

Related links

Project Management with Trello:
https://trello.com/b/LUkECqYG/final-project-ironhack

GitHub:
https://github.com/Beata2307/Facial_emotion_recognition_project