****

**COMP 6721 - Applied Artificial Intelligence (Fall 2023)**

**Project Assignment, Part 1**

**Team Name: AK\_17**

**Team Members: Alay Parikh (40269382)**

**Jenish Akhed (40270365)**

**Shruti Pavasiya (40270486)**

**Specializations:**

1. **Data Specialist: Shruti Pavasiya**
2. **Training Specialist: Jenish Akhed**
3. **Evaluation Specialist: Alay Parikh**

**GitHub Link:** [**https://github.com/parikhalay/Emotion-Detection**](https://github.com/parikhalay/Emotion-Detection)

Table of Contents

Dataset3

Overview of the Dataset3

Justification for dataset choices4

Provenance Information4

Data Cleaning6

Techniques and Methodologies6

Challenges7

Example Images8

Labeling10

Data Visualization11

Class Distribution11

Sample Images12

Pixel Intensity Distribution12

Reference Section14

**Dataset**

**Overview of the Dataset:**

The dataset used for this project is the "Affect Net Training Data" obtained from Kaggle[[1]](https://www.kaggle.com/datasets/noamsegal/affectnet-training-data), which contains a large collection of facial images categorized into different emotions.

Total number of images (including all classes) in existing dataset is: **28834**

There are 8 type of classes in the dataset, which are named below with the number of images in each class.

1. **Neutral Class:**

Total Images: 5127

1. **Surprise Class:**

Total Images: 4040

1. **Contempt Class:**

Total Images: 2872

1. **Anger Class:**

Total Images: 3219

1. **Disgust Class:**

Total Images: 2478

1. **Sad Class:**

Total Images: 3092

1. **Happy Class:**

Total Images: 5045

1. **Fear Class:**

Total Images: 2960

**Special Characteristics of the Dataset :**

1. Diverse Emotions: The dataset includes a wide range of emotions, making it suitable for training a model to recognize not only basic emotions but also nuanced expressions. Images are captured from all front angles.
2. Real-World Variety: The images in the dataset feature individuals from various demographics, making it applicable to real-world scenarios.
3. Varied Lighting and Backgrounds: The images may have diverse lighting conditions and backgrounds, which adds to the complexity of emotion recognition.

**Justification for Dataset Choices:**

1. **Relevance to the Project:** The dataset is highly relevant to the project's objective, which is to develop an AI system for analysing and categorizing student expressions in an academic setting. Emotion recognition is a key component of this system.
2. **Rich Emotion Coverage:** The dataset provides a comprehensive collection of emotions, enabling the model to recognize a broad spectrum of emotional states.
3. **Real-World Applicability:** The dataset's diversity in terms of subjects, expressions, and conditions aligns with the real-world scenarios in which the AI system will be deployed.

**Challenges:**

1. **Data Size:** The dataset is extensive, which can be both an advantage and a challenge. Training a model on a large dataset requires significant computational resources.
2. **Class Imbalance:** While the dataset covers a variety of emotions, there may be class imbalance issues, with some emotions having more examples than others.
3. **Data Pre-processing:** Due to the dataset's diversity, pre-processing steps such as resizing and data augmentation are required to ensure uniformity for model training.

**Provenance Information:**

The "AffectNet Training Data" dataset was sourced from Kaggle, and it is provided by Noam Segal. The dataset's original page on Kaggle contains information regarding its origin, licensing, and other relevant details. Below is a summary of the provenance information:

|  |  |  |
| --- | --- | --- |
| **Image Class** | **Licencing Type** | **Comments** |
| Kaggle (Noam Segal) | Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO) | [Kaggle Dataset] |
| Mollahosseini et al., 2017 | N/A | Affect Net dataset |
| Affect Net Research Project (2017) | N/A | Original source of the dataset |

**Data Cleaning**

**Techniques and Methodologies:**

**File Format:** Conversion from .png file format to .jpg file format for uniform data visualization. Following are the files used to achieve that process:

* **pngTojpg.py:** This script is designed to convert PNG image files to JPG format. It iterates through directories, identifies PNG files, converts them to JPG, and replaces the original PNG files[2].
* **datasetShrinking.py:** This script selects and copies a specific number of random images from a source directory to a target directory. It's useful for creating smaller subsets of a dataset for testing and experimentation[2].

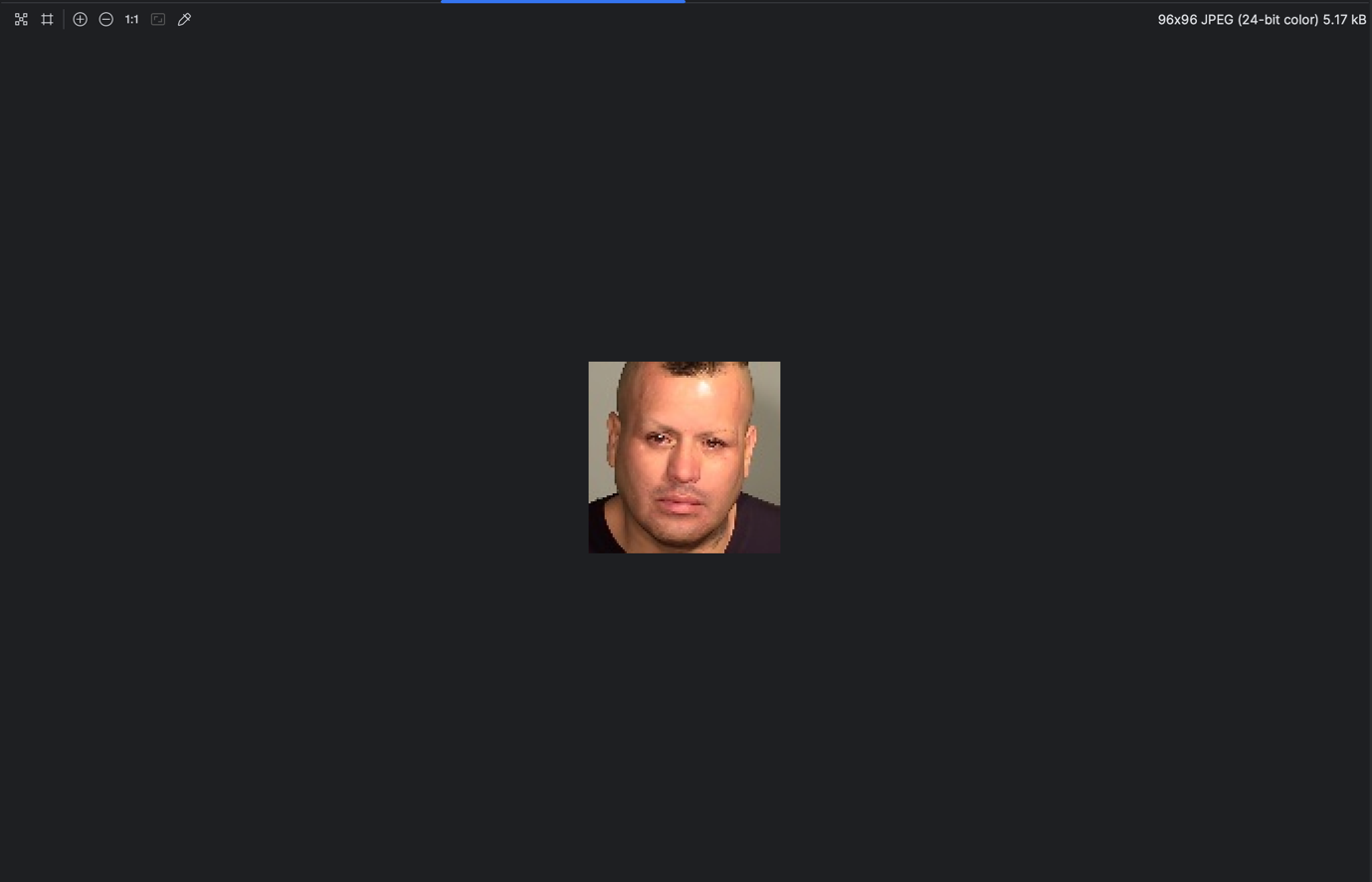
1. **Resizing Images:**
   * To standardize the dataset, the input images are resized to a consistent dimension. The target dimensions for resizing are set to a width of 100 pixels and a height of 100 pixels (target\_width = 100 and target\_height = 100).
   * The resizing process preserves the aspect ratio to ensure that the image's proportions remain intact. If the original width is greater than the original height, the width is set to the target width, and the height is calculated accordingly. Conversely, if the original height is greater, the height is set to the target height, and the width is calculated accordingly.
   * The interpolation method used for resizing is cv2.INTER\_LANCZOS4, which is a high-quality interpolation method that helps avoid pixel loss and maintain image quality during resizing[3].
2. **Brightness Adjustment:**
   * A brightness adjustment is applied to the images. The brightness factor is set to 1.1 (brightness\_factor = 1.1), which slightly increases the overall brightness of the image.
   * The adjustment is applied after resizing and pasting the image onto a canvas, ensuring that the entire image, including the canvas area, is adjusted.

**Challenges Encountered and Addressed:**

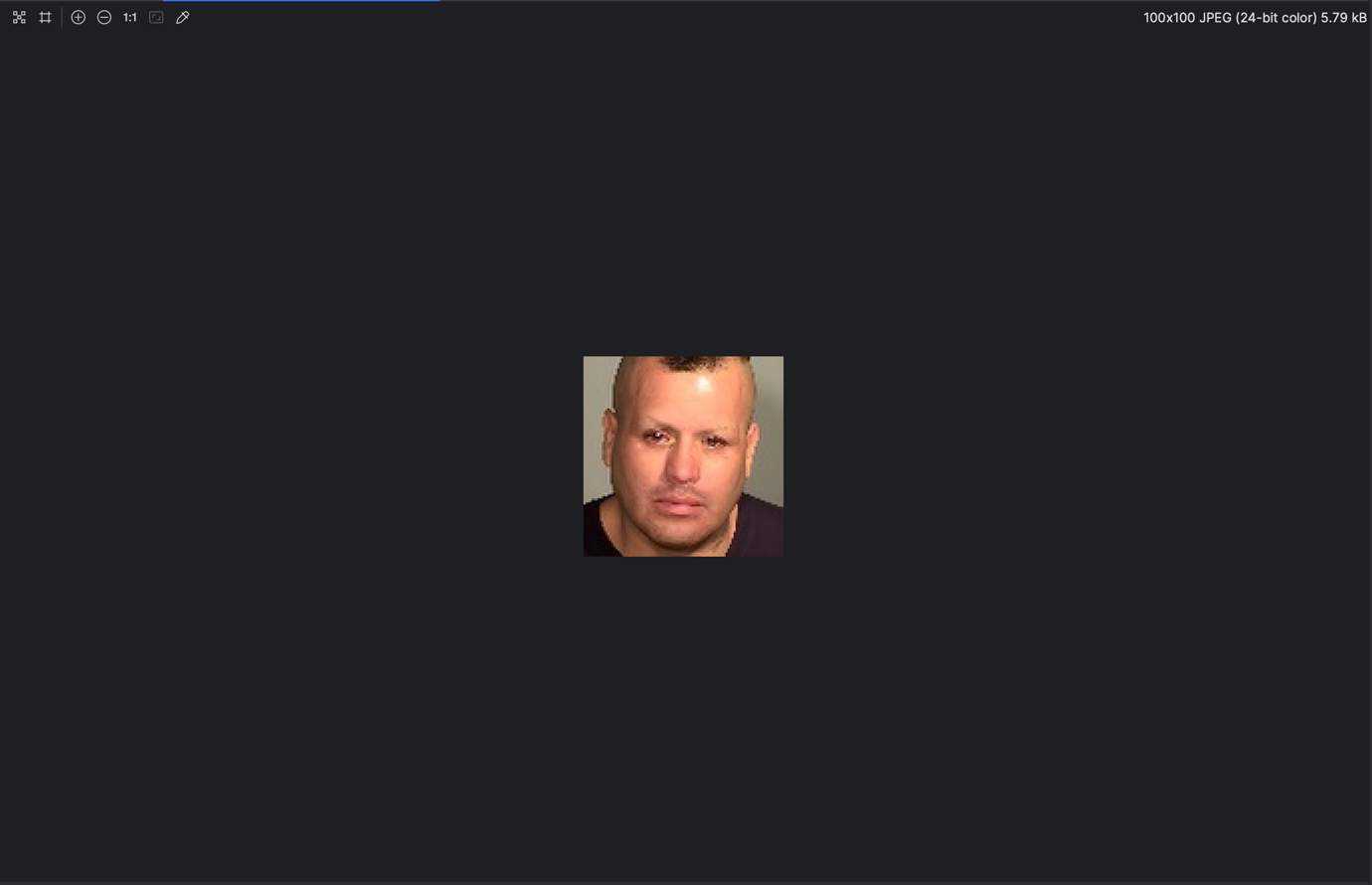
* **Aspect Ratio Preservation:** One challenge during resizing was preserving the aspect ratio of the original images. This was addressed by calculating the new dimensions based on the aspect ratio, ensuring that the images did not appear distorted.
* **Brightness Adjustment:** While applying brightness adjustments, it was important to ensure that the entire image, including the canvas area, was adjusted consistently. This was addressed by applying the brightness adjustment after the image was pasted onto the canvas.
* **Pixel Loss During Resizing:** To prevent pixel loss and maintain image quality, the interpolation method **cv2.INTER\_LANCZOS4** was chosen for resizing. This high-quality interpolation method results in smoother and sharper images after resizing.

Examples of the Before-and-After Cleaning Effect:

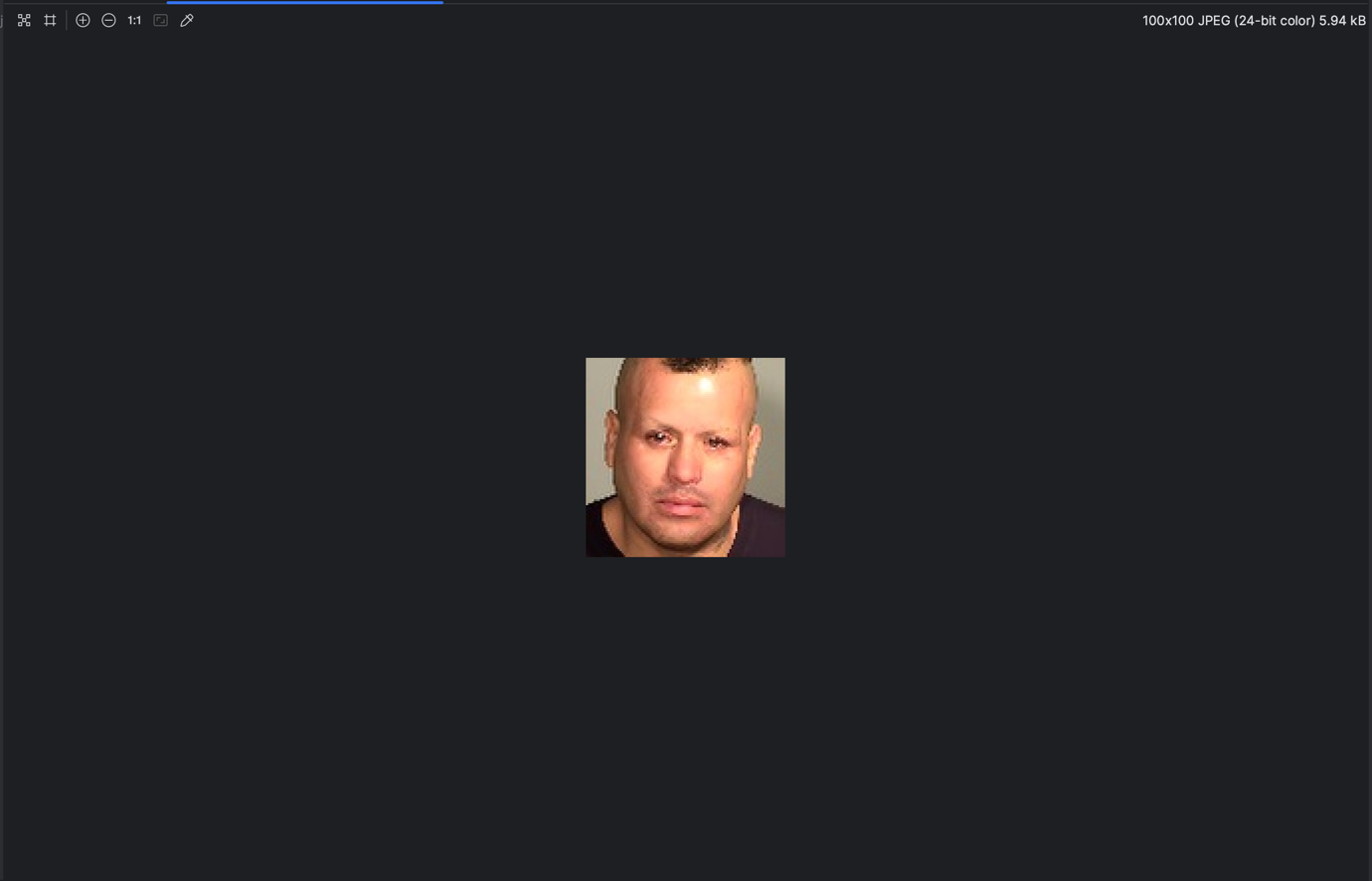
Original Image:



Resized Image with Aspect Ratio Preservation:



Resized Image with Brightness Adjustment:



In the examples, the original image is first resized while preserving its aspect ratio. The resized image is then pasted onto a canvas of the target dimensions. After pasting, a brightness adjustment is applied, enhancing the overall brightness of the image.

The data cleaning process ensures that all images are consistent in size and appearance, making them suitable for subsequent data analysis and model training.

**Labelling**

Dataset labelled in reference dataset:

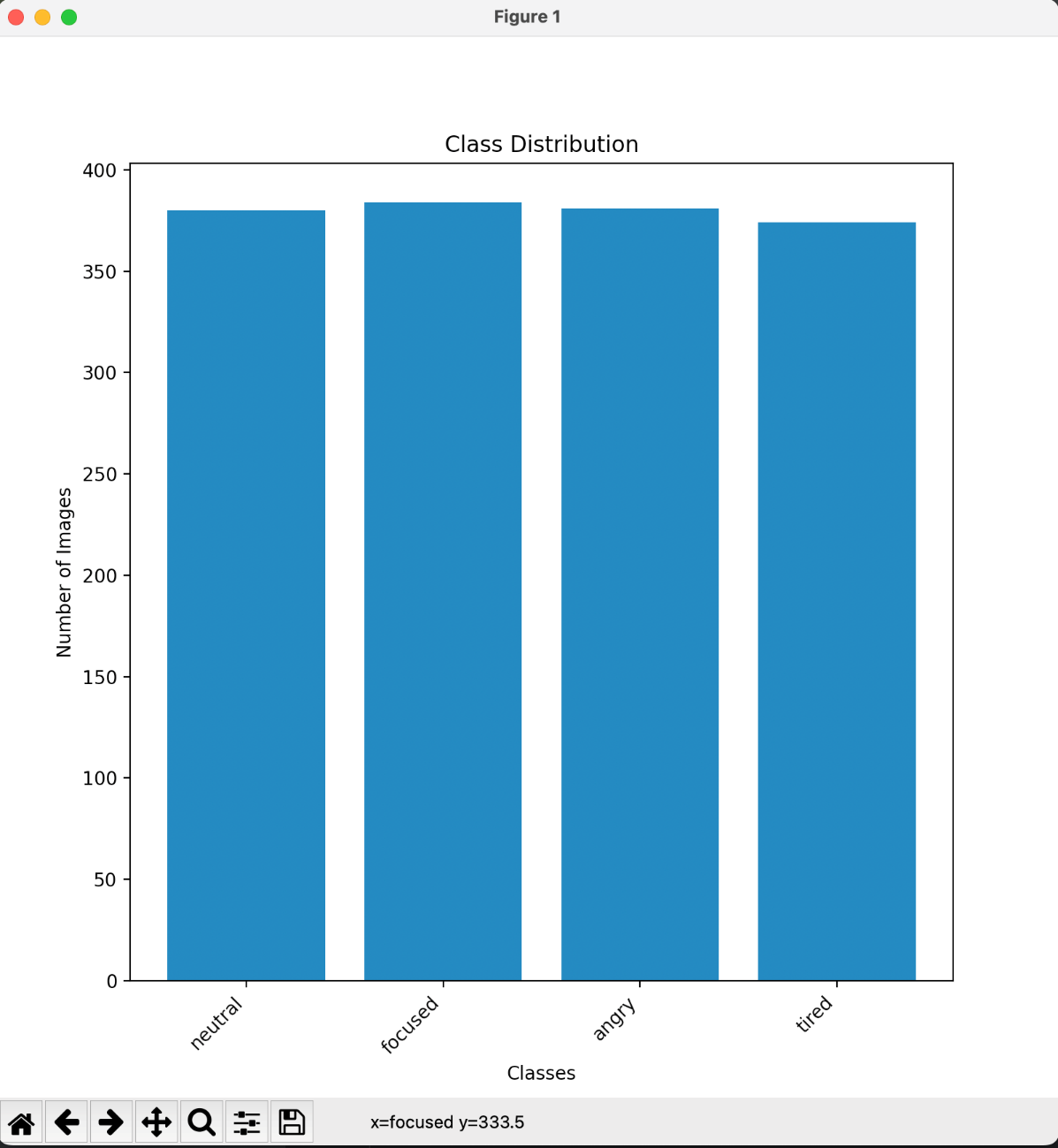
* The dataset was created on Label box[4], a data labelling platform.
* It is expected to be used for facial expression analysis, including emotions such as neutral, engaged, bored, angry, confused, and distracted.
* The Affect Net dataset was already labelled into its respective classes, so there was no need for further labelling and analysing with label box as it is already labelled with label box tool.

**Data Visualization**

**1) Class Distribution(Bar Graph)[5]:**

* The bar graph below shows the number of images in each class.
* It provides insights into whether any class is overrepresented or underrepresented.
* The class labels are listed on the x-axis, and the number of images is on the y-axis.

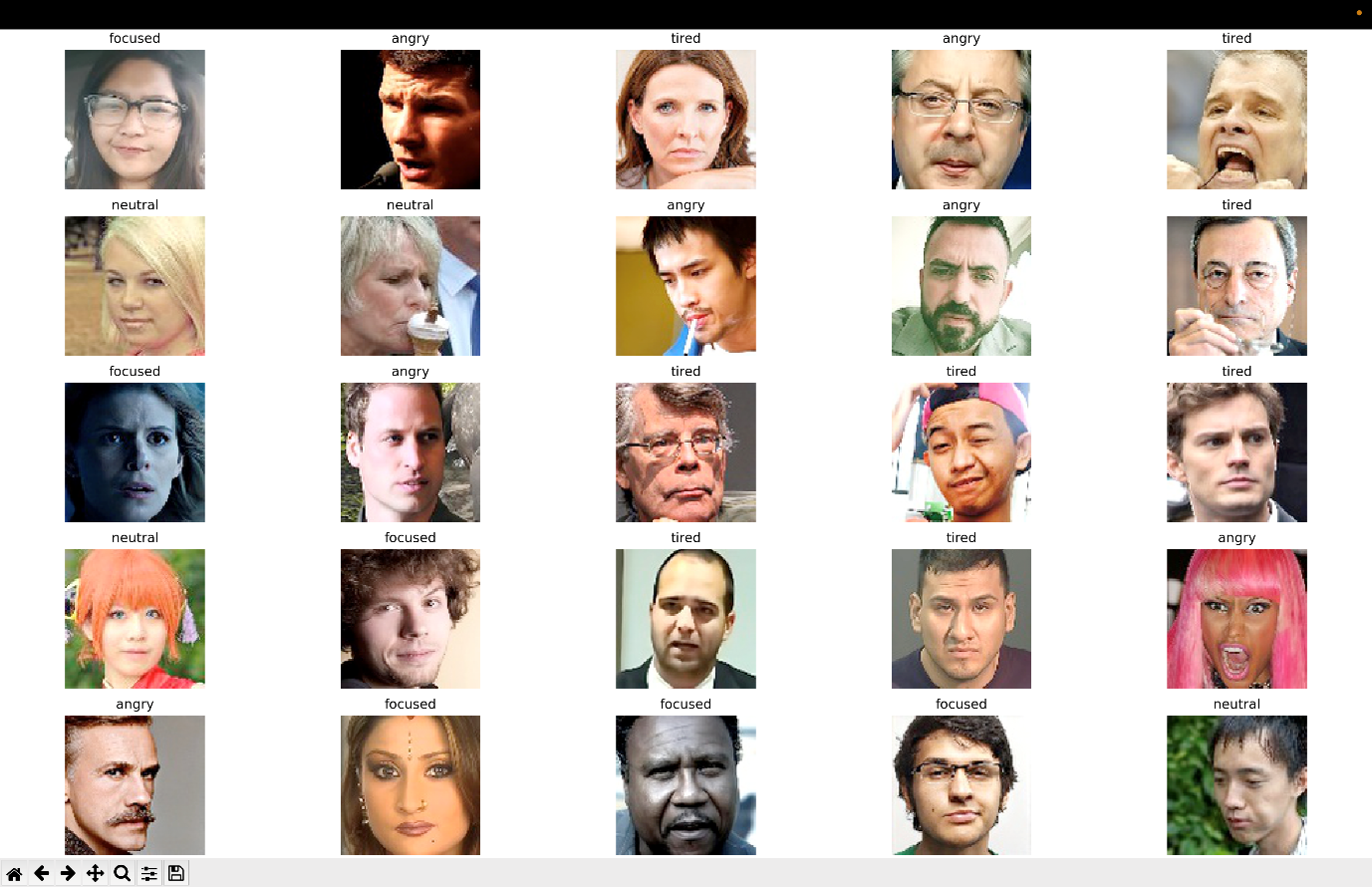
Example:



**2) Sample Images(5x5 Grid)[6]:**

* A collection of 25 random images is presented in a 5x5 grid.
* The images are randomly chosen from different classes upon every code execution.
* This visual representation helps understand the dataset's content and may help identify any noticeable anomalies or potential mislabelling’s.

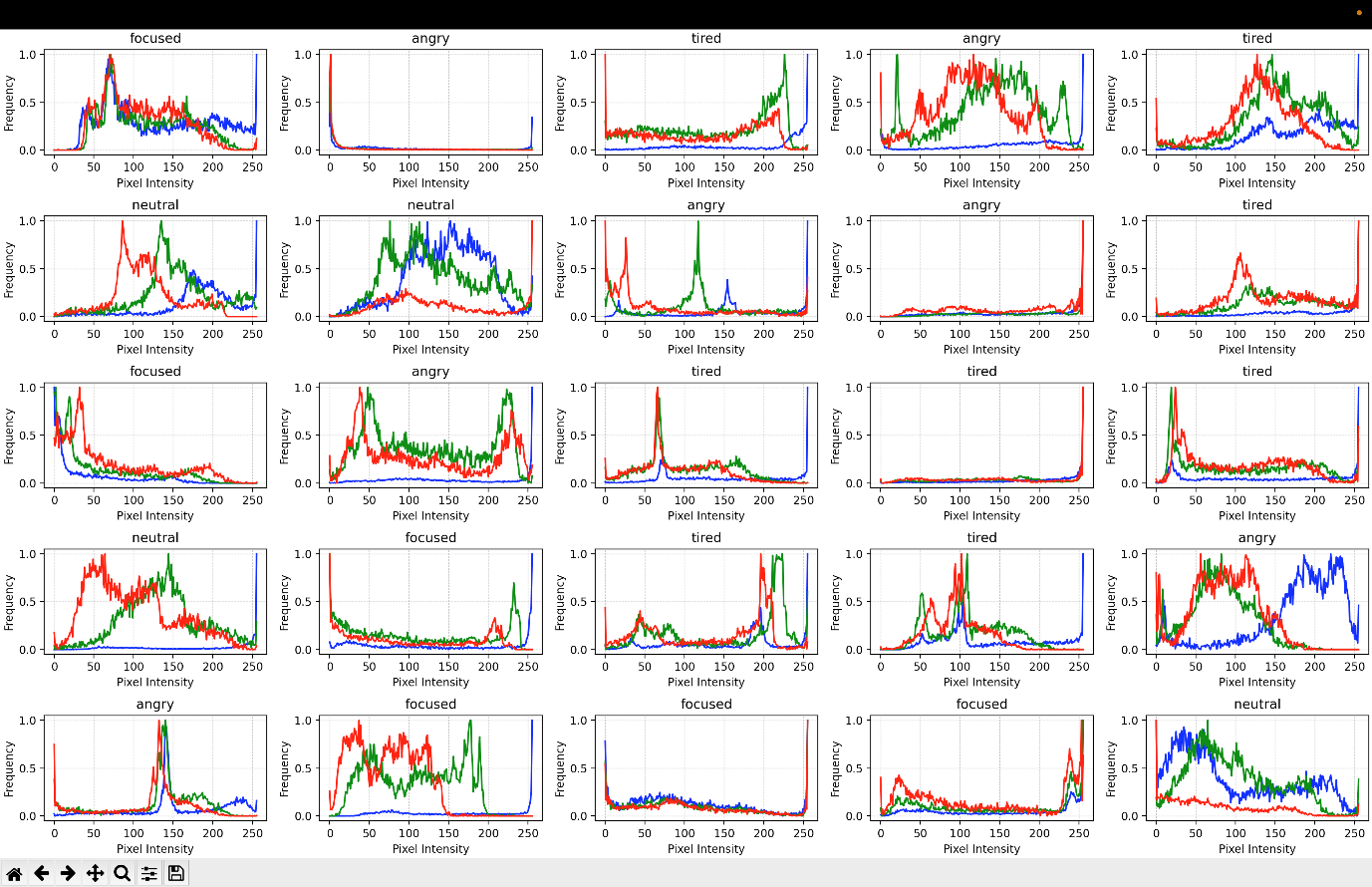
Example:



**3) Pixel Intensity Distribution[6][7][8]:**

* For the same random images, histograms showing the distribution of pixel intensities are plotted.
* In the histogram, the pixel intensity distribution of the Red, Green, and Blue channels is overlaid on a single histogram.
* This provides insights into variations in lighting conditions among the images.

Example:



**References**

[1] Dataset: <https://www.kaggle.com/datasets/noamsegal/affectnet-training-data>

[2] For OS related code: <https://docs.python.org/3/library/os.html>

[3] Resize Image: <https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html>

[4] Label Box - <https://labelbox.com/>

[5] Bar Graph: <https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html>

[6] Subplots: <https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.subplots.html>

[7] Histogram Calculator: <https://docs.opencv.org/3.4/d8/dbc/tutorial_histogram_calculation.html>

[8] ChatGPT - <https://chat.openai.com/>

[9] Image Histogram Checker: <https://www.sisik.eu/hist>

[10] Python - <https://www.python.org/>