Gina Cody School of Engineering and Computer Science
Concordia University

COMP 6721 – Artificial Intelligence
Project Assignment Part-1

SNO	Student Name	Student ID	Specialisation
1	Roshini Chukkapalli	40198936	Data Specialist
2	Srujitha Yeruva	40232527	Training Specialist
3	Sree Sneha Kothapalli	40235510	Evaluation Specialist

Link to GitHub Repository: https://github.com/rochuk/COMP-6721---AI-ducation-analysis-project

Dataset:

The objective of this project is to develop a Deep Learning Convolutional network using PyTorch. With this objective, we can analyse the images of students or people and categorize them into various classes. To achieve this objective, the classes we considered in this project are Neutral, Engaged/Focused, Bored/ Tired and Angry / Irritated.

Overview of the datasets:

The total number of images present in the dataset are 2680.

Neutral:

This facial expression is recognised when his face is usually relaxed and has no other emotional expression involved. There are a total of 750 images for this class including the training and test data. The training data has 625 images, and the test data has 125 images in total. Most of these images are frontal face shots and the images are of size 224*224 pixel. These images are in PNG format and this class contain images that are from various backgrounds. All these images are not uniform images and has still poses. The images in this class are squared images and focusses mainly on the facial expression of the person.

Engaged/Focused:

This facial expression is recognized when a student's face shows evidencing signs of active concentration, with sharp and attentive eyes. There are total of 550 images for this class including training and test data. Most of these images are frontal face shots and the images are of size 224*224 pixel. These images are in PNG format and this class contain images that are from various backgrounds. All these images are not uniform images and have still poses. The images in this class are squared images and focus mainly on the facial expression of the person.

Bored/Tired:

This facial expression is recognised when a person exhibiting signs of fatigue or disinterest, which can manifest through tired-looking eyes or absent-minded gazes. The dataset comprises 630 PNG images of excellent quality including training and test data. Most of them having a resolution of 1024x1024 pixels. It exhibits significant diversity in terms of age, ethnicity, and background settings, ensuring a broad representation of human characteristics. Additionally, the dataset provides comprehensive coverage of various accessories like eyeglasses, sunglasses, hats, and more. These images were obtained by web crawling from Flickr, which means they may reflect the biases present on that platform.

Angry/Irritated:

This facial expression is recognised when his face has signs of agitation or displeasure which may result tightened facial muscles, narrow eyes, and a tight-lipped frown. There are a total of 750 images for this class including the training and test data. The training data has 625 images, and the test data has 125 images in total. Most of these images are frontal face shots and the are of size 224*224 pixel. These images are in PNG format and this class contain images of people from various backgrounds. All these images are not uniform images and has still poses. The images in this class are squared images and focusses mainly on the facial expression of the person.

Justification:

Neutral and Angry:

These classes are particularly taken from Kaggle. The reason for considering this dataset in Kaggle is because, these datasets include a wide variety of frontal shots and images with diverse backgrounds. On top of that, these images are classified images which reduces the work to manually sort them. These images do not contain duplicate images and are of good quality. This dataset also has a greater number of images in each class which allows us to choose the images we want as our requirement should be a minimum of 2000 images in total. Although some images have more pixels and cannot be visualised properly, it has been taken care of by using light augmentation techniques.

Engaged:

This dataset is a combination of images obtained from multiple sources on the internet. We have manually picked the relevant images from the kaggle dataset and istock (an online image library) mentioned in the source in below section, as we couldn’t find a direct dataset. As we have picked the images manually from the mentioned sources, some images had to be converted into gray scale images and resized to uniform size. The images have been processed, including alignment and cropping, using the PIL, dlib libraries.

Bored:

The choice of datasets was made to ensure that we have a diverse and balanced dataset for our project, which aims to classify facial expressions into different emotional states. The "Bored" class is particularly relevant as it represents one of the mandatory emotions to be detected. The "Bored" images were selected from the Flickr-Faces-HQ (FFHQ) dataset, a comprehensive dataset of high-quality human faces. We chose this dataset because it provides a wide variety of facial expressions, including expressions of boredom. The challenge we encountered was in accurately labeling images with the correct class due to differences in class definitions among the source datasets. Furthermore, the images have been automatically processed, including alignment and cropping, using the dlib library.

Detailed information about images:

SNO	Class	Source Link of dataset	AuthorName	Licence
1.	Neutral	https://www.kaggle.com/datasets/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef/data	Sudarshan Vaidya (Owner)	Unknown
2.	Angry	https://www.kaggle.com/datasets/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef/data	Sudarshan Vaidya (Owner)	Unknown
3.	Bored	https://github.com/NVlabs/ffhq-dataset	Tero Karras (NVIDIA), Samuli Laine (NVIDIA), Timo Aila (NVIDIA)	Flickr
4.	Engaged	https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer https://www.kaggle.com/code/hanimohamed/student-engagement-model/input istock	ARES Hani Mohamed	CC0: Public Domain

Data Cleaning:

Data Cleaning is a process on which we remove or fix duplicates, incorrectly formatted, orientation of the dataset. This step is very crucial step to build a convoluted neural network. Here, in this project, we are collecting datasets from multiples sources so there are high chances that the data might be labelled incorrectly or may contain duplicate images. If this happens, then there are chances that the algorithm may provide incorrect or biased results. So, data cleaning is a very necessary step.

The images can have varied sizes, resolutions, and lighting issues. To make sure that the images were of same size and more robustness, in this project, after collecting the data from the dataset, a python script is written to clean the data.

Resizing the images:

All the images in the dataset have a constant size i.e., 256*256 pixels. This is performed by defining a transformation, a function transforms.resize() in which the size of the images are are passed as parameters. This is to ensure that the model is trained properly and gives the exact output.

Greyscale conversion:

All the images in the dataset are converted to grey scale so that the facial expression of the dataset is more precise and can be clearly predicted. This is achieved by using the transforms library. The function called is transforms.ToPILImage().

Light augmentation:

All the images in the dataset are light augmented. They are adjusted to a brightness factor of 0.2 and the contrast factor is 0.2. This is achieved by the transforms library. The function called is tranforms.ColorJitter().

After the data cleaning process is done, all the images are saved into a separate directory.

In the below figures, ‘a’ represents the original image whereas ‘b’ represents the data cleaned image with size being increased to 256*256, converted to a grey scale and has improved light i.e. brightness and contrast of the image is improved.

a. Original Image

b. Data Cleaned Image.

Removing duplicates:

If there are any duplicate images among all the categories of datasets their labels are displayed after running the code and the method used for this purpose is "remove_duplicates(image_directory)". It maintains a dictionary image_hashes to store hash values of processed images, where the keys are the hash values, and the values are the image file paths. It initializes an empty list duplicate_images to keep track of images that are identified as duplicates. It iterates through the files in the image_directory using os.listdir(image_directory). For each file, it checks if the filename ends with ".jpg" to filter out non-image files and it calculates the image hash using the get_image_hash function. If the hash is not already in the image_hashes dictionary, it adds the hash and image file path to the dictionary. If the hash is already in the dictionary, it identifies the image as a duplicate and appends its file path to the duplicate_images list. After identifying all duplicate images, it iterates over duplicate_images and removes them from the directory using os.remove(duplicate_image). Finally, it prints a message for each removed duplicate image, indicating its removal.

Labeling:

The process of labeling our dataset involved multiple steps, including the use of tools and manual verification to ensure accurate and consistent class assignments. Methods, tools, and challenges faced during the labeling process are described below.

A combination of automated class assignment and manual verification to label the images in our dataset were used.

1. Initial Class Assignment: Initially, we leveraged metadata and file-naming conventions to automatically assign class labels to the images. The class details embedded in the file names allowed us to categorize the images into distinct groups such as "Angry," "Neutral," "Engaged," and "Bored," relying on numeric prefixes.
2. Hands-On Verification: To ensure the accuracy of class assignments, we carried out manual validation. This process involved a visual inspection of each image to confirm that the assigned class label faithfully represented the subject's facial expression. Images presenting any doubts or inconsistencies were identified for further in-depth review.

During the labeling process, we encountered a few ambiguities and challenges that required careful consideration:

1. Unclear Facial Expressions: In certain instances, facial expressions did not neatly fit into a single emotional category. For example, an image might exhibit a combination of "Neutral" and "Engaged" expressions. In such scenarios, we engaged in internal deliberations among team members to arrive at a consensus regarding the most suitable label.
2. Discrepancies in Source Datasets: Our dataset resulted from the amalgamation of images sourced from diverse datasets, each with its own class definitions, occasionally diverging from our intended classes. To address this, we devised a set of mapping rules to ensure that images originating from various sources were correctly assigned to the appropriate classes for our project.
3. Differences in Illumination Settings: Images sourced from various origins displayed variances in lighting conditions, which had the potential to influence the interpretation of

facial expressions. We took these lighting disparities into consideration when assigning class labels, while also considering the broader context of the image.

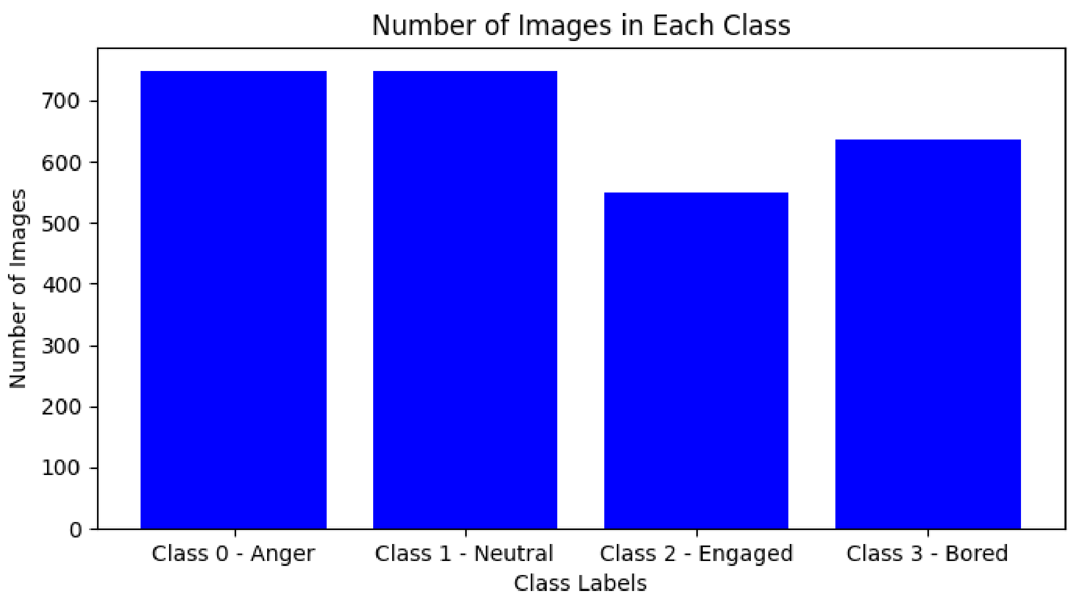
The entire labeling process was iterative and involved multiple rounds of review and discussion to ensure the accuracy and consistency of class assignments. The final dataset was thoroughly validated to reflect the intended labeling for our facial expression classification project.

Dataset Visualization:

We employ Matplotlib to visualize our dataset, providing insights into its class distribution, showcasing sample images, and examining pixel intensity distributions.

Class Distribution:

To begin, we present a bar graph illustrating the distribution of images across our four classes: "Angry," "Neutral," "Engaged," and "Bored." This analysis is crucial to assess if any class is overrepresented or underrepresented in our dataset, as imbalanced class distributions can impact model performance.



The class distribution graph reveals that our dataset is well-balanced, with each class containing an approximately equal number of images. This balanced distribution ensures that our machine learning model is exposed to a diverse range of facial expressions, facilitating effective training.

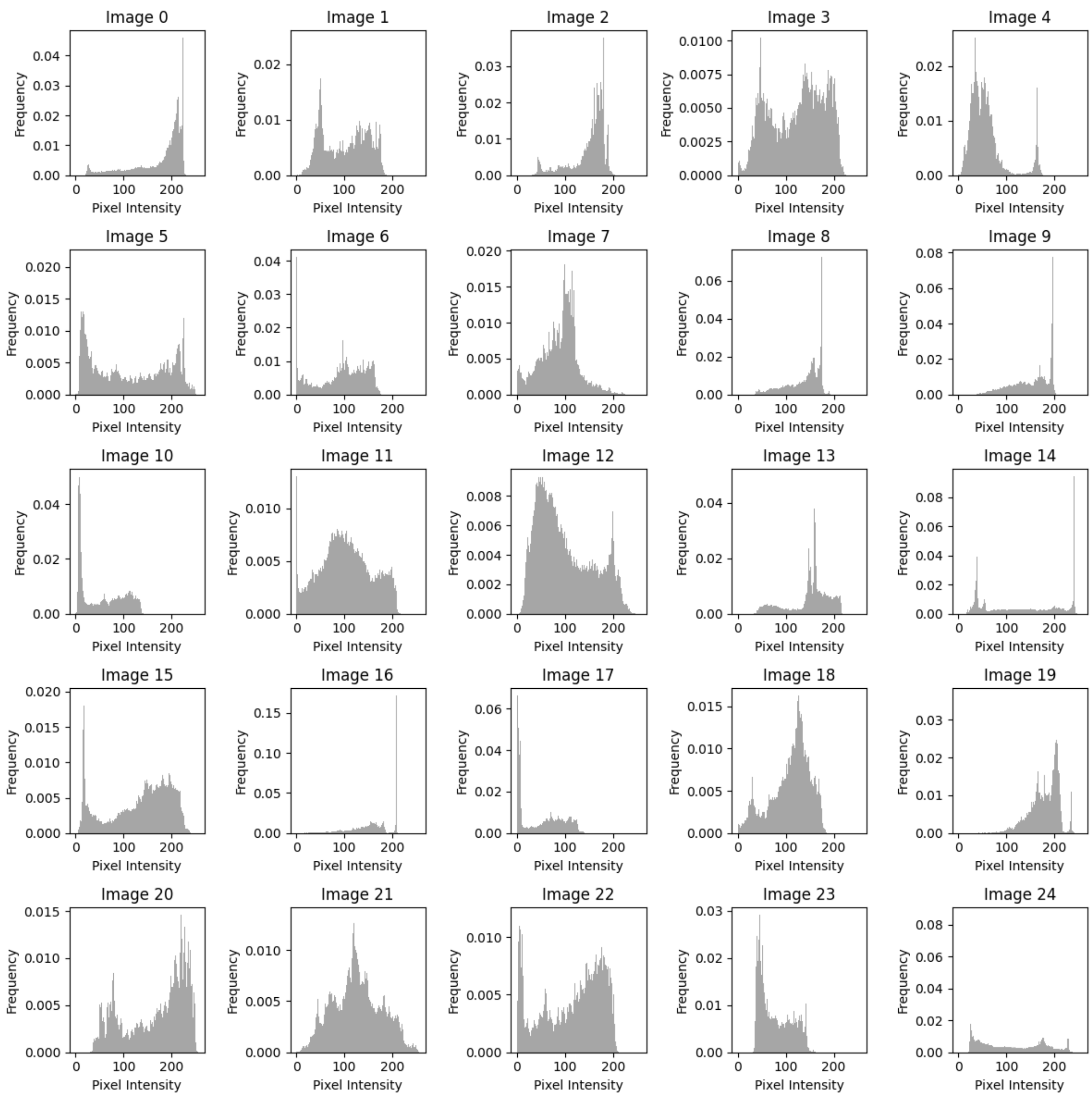
Sample Images

Next, we showcase a collection of 25 sample images organized in a 5x5 grid. These images are randomly selected from each class, ensuring that the selection varies with each code execution.

The sample images provide a visual representation of our dataset's content, allowing us to gain a deeper understanding of the facial expressions captured. This visual inspection also helps us identify any potential anomalies or mislabelings in the dataset, ensuring data quality.

Pixel Intensity Distribution

To delve further into the dataset's characteristics, we analyze the pixel intensity distribution of the same random images. The pixel intensity distribution reveals variations in lighting conditions among the images.



In the histogram, we overlay the intensity distributions of the Red, Green, and Blue channels for color (RGB) images. Analyzing these distributions aids in identifying variations in lighting conditions, which is essential for understanding the dataset's diversity.

Overall, this dataset visualization process provides valuable insights into the dataset's composition, balance, and quality. It ensures that we have a well-structured and representative dataset for our facial expression classification project.

References:

1.https://www.tableau.com/learn/articles/what-is-data-cleaning#:~:text=Data%20cleaning%20is%20the%20process,to%20be%20duplicated%20or%20mislabeled.