SmartClass A.I.ssistant

Project Assignment: Part I

Due: Friday, May 31, 2024

Team name: FS-4

Team members:

Tristan Manni-Mirolla (40112168); GitHub: TristanMM2; Training Specialist

Alin Caia (40246955); GitHub: SerbanCaia; Evaluation Specialist

Melissa Ananian (40112159); GitHub: Meliimoon; Data Specialist

GitHub Project Repository: https://github.com/serbancaia/ai-comp472

We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality

Signed: Tristan Manni-Mirolla (40112168), Alin Caia (40246955), Melissa Ananian (40112159)

Date: Saturday, May 25, 2024

Dataset

For our project, we chose to train our AI model on four emotions: Neutral, Angry, Focused, and Happy. To compile our own datasets for training and testing our AI model, we handpicked photos from three different datasets found online. The first dataset [1], has a total of 29,000 images and 8 classes, with each class containing around 2000-5000 images sized 96x96px. This dataset contains colored headshots of individuals with some imagery still visible in the background. Dataset [2] has a total of 71,800 images and 7 classes, with each class containing around 500-7000 images sized 48x48px. Dataset [3] has a total of 35,900 images and 7 classes, with each class containing around 400-7000 images sized 48x48px. Datasets [2] and [3] are similar in that they contain greyscale images of mostly zoomed in headshots with minimal visible backgrounds. All three datasets have face shots from multiple different angles for all the expression classes.

These datasets were chosen due to their extensive collection of images and the diversity of image subjects within them. The datasets all contain images of subjects of different genders, races, and ages. The images also vary in their lighting, intensities of the facial expressions, and angles the subjects are faced. These details are relevant to the training of our AI model as it will help the model learn to recognize human expressions from unique faces under various and broader conditions. When picking images from these datasets to use in our own, we made sure to include a representative amount of these varying features to minimize any biases and ensure a more robust model.

The challenges faced by these datasets were caused by a few factors. Firstly, datasets [2] and [3] had some overlapping images, therefore we had to be extra vigilant in picking our images as to not accidentally have duplicates in our dataset's classes. Second, the datasets were organized into labeled classes of emotions, however, we found that there were many instances of mislabeling. For example, when looking through the 'Neutral' dataset, there were few images where the subject was clearly displaying an excited expression, again pushing us to be more careful in the analyzation of these images when choosing ones to include in our dataset. Third, there was the challenge of ambiguous expressions. Some facial expressions we encountered in the datasets were ambiguous and we felt they could have fit in multiple classes. In these cases, we opted to include some images in more than one class. Lastly, the datasets we chose did not have a class for the Focused expression. For this reason, we had to manually compile the images for the Focused class using the ambiguous images seen in other classes of the datasets we were using. Some images from the Neutral, Angry, Contempt, Sad, and Happy classes had expressions that could have also been interpreted as being a focused look. Thus, through reviewing those classes' folders, we compiled the necessary number of images for our Focused class training and testing datasets.

Provenance Information			
Directory	**Sub-directory**	**Image Batch**	**Referencing**
Training	**Angry**	custom_angry1.jpg, custom_angry2.jpg, custom_angry3.jpg	These images are personal images taken by each team member
		image0000006.jpg – image0024998.jpg	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
	Focused	custom_focused1.jpg, custom_focused2.jpg, custom_focused3.jpg	These images are personal images taken by each team member
		65.jpg – 33361.jpg	Reference List: [2]
		ffhq_136.png – ffhq_5637.png, image0001555.jpg – image0042251.jpg	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		Training_4175225.jpg – Training_99766691.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents
	Happy	custom_happy1.jpg, custom_happy2.jpg, custom_happy3.jpg	These images are personal images taken by each team member
		8.jpg – 21384.jpg	Reference List: [2]
		ffhq_0.png – ffhq_1039.png	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		Training_123362.jpg – Training_35877462.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents
	Neutral	custom_neutral1.jpg, custom_neutral2.jpg, custom_neutral3.jpg	These images are personal images taken by each team member
		4.jpg – 15140.jpg	Reference List: [2]
		ffhq_0.png – ffhq_2030.png	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		Training_65667.jpg – Training_99766691.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents

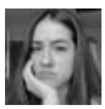
Testing	**Angry**	23.jpg – 4077.jpg	Reference List: [2]
	Focused	298.jpg – 35764.jpg	Reference List: [2]
		ffhq_4311.png – ffhq_5644.png, image0022384.jpg – image0039941.jpg	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		PrivateTest_1129340.jpg – PrivateTest_85251764.jpg, PublicTest_7059610.jpg – PublicTest_64856942.jpg, Training_89691211.jpg – Training_99111610.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents
	Happy	22340.jpg – 30779.jpg	Reference List: [2]
		ffhq_3829.png – ffhq_4987.png, image0031576.jpg – image0039941.jpg	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		Training_35762019.jpg – Training_99762935.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents
	Neutral	44.jpg – 35764.jpg	Reference List: [2]
		ffhq_4307.png – ffhq_5659.png, image0022379.jpg – image0041848.jpg	Reference List: [1] Licensing type: Attribution-NonCommercial-ShareAlike 3.0 IGO (CC BY-NC-SA 3.0 IGO)
		PrivateTest_932822.jpg – PrivateTest_85251764.jpg, PublicTest_7059610.jpg – PublicTest_83833485.jpg	Reference List: [3] Licensing Type: Database: Open Database, Contents: Database Contents

Data Cleaning

Due to the size and color discrepancies of the images found across the three selected datasets, we opted to standardize the data through two methods: resizing each image to 48x48 pixels and converting each image to grayscale (using the 'resizing.py' script that we wrote), as seen in the table below. We decided on the size 48x48 because two of the three datasets had their images set to this size, therefore we resized the third dataset's images from their original 96x96 size to our 48x48 standard. This decision was made because upscaling the 48x48 images to 96x96 would have resulted in lower resolution and pixel density, thus negatively impacting the quality of the images that our model will eventually train on. Furthermore, converting the images to grayscale allowed our scripts to run faster, since the color value was only stored in one channel with values from 0-255, instead of 3 channels (R, G, B) as is the case for RGB images. This caused the grayscale 48x48 images to be flattened to a 1-D array of size 2304 (with each index corresponding to a pixel in the image) rather than a 1-D array of size 6912. As mentioned above, the script 'resizing.py' that we wrote was used to resize and grayscale each dataset class. This script took the absolute path containing the dataset as input and iterated through every file in the directory, using the Pillow library [4], [5] to open, resize, and convert to grayscale, before saving the image.

One challenge that we encountered during the data cleaning process was ensuring that there were no duplicate images within our dataset classes or between the training and testing datasets. Unfortunately, many of the data appeared across datasets. Since the images were manually selected from these online datasets [1], [2], [3], it was difficult to guarantee that we had not mistakenly selected a duplicate image that had already been chosen from another dataset containing the same image. To address this concern, we wrote 'dupecheck.py' which, with the help of OpenCV [6], took an absolute path as input, and checked every image with every other image in the directory to see if there were any duplicates. The logic behind this script was that every 48x48 image was flattened to a 1-D contiguous array of size 2304 (with the .ravel() method [7]), with each index containing a value between 0-255, corresponding to its pixel intensity, and then two images were checked for duplicates by comparing every corresponding index of the two arrays. If any value at a given index did not match the other array's value at that same index, then that meant that that given pixel was different, thus meaning that the images were not the same. Conversely, if every value at every index was the same for both images, this meant that they were the same image and would be flagged as a duplicate to be automatically deleted from the directory. Once we had finalized our datasets, we ran the script on them and found that only 0.7% of the images were duplicates and were subsequently deleted.

Ultimately, the decisions we made to standardize the data in the way that we did simplified and improved certain aspects of the data cleaning process, such as the performance of the 'dupecheck.py' script. Aside from the negative impact to image resolution, if we had resized the 48x48 datasets to 96x96 and kept the images as RGB, the length of the 1-D array representing pixel intensities would have increased from 2304 to 27,648, becoming 12 times larger and thus increasing the runtime of our scripts used for data cleaning. Other challenges, such as ambiguous or mislabeled data required more manual solutions, which will be discussed in the Labeling section of this report.

Original Image	Standardized Image	Original Image	Standardized Image
 (custom_angry1.jpg)		 (ffhq_1032.png)	
 (custom_angry2.jpg)		 (ffhq_420.png)	
 (custom_happy3.jpg)		 (6325.jpg)	no change

Labeling

The platform we used to retrieve the images in our datasets was Kaggle. We decided not to use other platforms such as Labelbox for the first part of the project as we did not feel the need to use them. However, we may do so during the upcoming parts. We examined a few available facial expression recognition datasets posted on Kaggle and picked images from multiple of those datasets. The classes we chose were Angry, Focused, Happy, and Neutral. We took inspiration from our professor and the Kaggle datasets and decided that it would be simple to assign labels to images by inserting them in the folder named after their appropriate label. We manually selected the images we found from the datasets on Kaggle based on how appropriate they looked for our project.

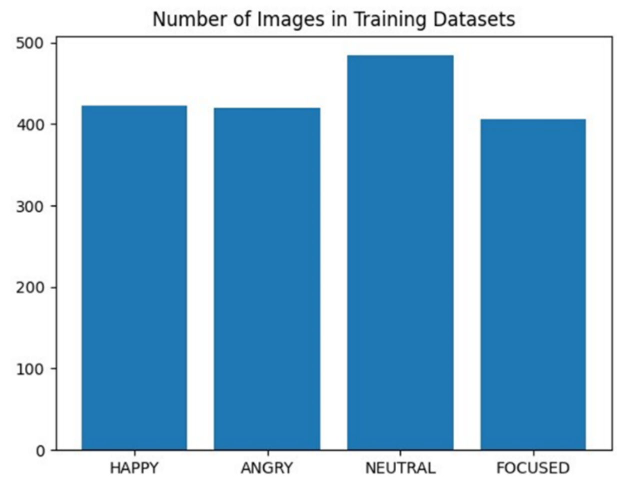
There were four major challenges we faced when it came to labeling. The first was the lack of datasets online that have a Focused label, leaving us without any actual examples of data that can be labeled that facial expression. The second was the ambiguity of how a focused person's face may resemble since the spectrum can span the Angry, Happy, and Neutral classes. The third concerned how our A.I. would perceive students in a large classroom. Since the angle and point of view of the A.I. could differ based on how many cameras are installed in the classroom, which locations they are installed at, and which direction students must face to pay attention to the learning material, the training data may require images of individuals looking towards various directions instead of only camera-facing snapshots. The fourth had to do with the fact that some of the images we found in the online datasets were either showing fake people or were doctored so that the expression they showed matched the label. Thankfully, these pictures were not too difficult to identify. Thus, we were able to manually include ones that were neither fake nor doctored.

To solve the third issue, since the scope of this project includes both in-person and online learning settings, we decided to assume that the A.I. would be using one camera to view the entire class of students since it would be the most practical way of implementing such a technology in both real-world settings, meaning that it would observe the facial expressions of students at different angles. This decision impacted the way we classified the data we retrieved, especially for the Focused class. The reason for this is because, if the direction that a person faces in the picture varies, then this feature becomes less precise, which would decrease its weight value when training the A.I. model. Therefore, most of the critical inputs during classification would relate to an individual's facial structure more than the direction towards which the eyes are facing.

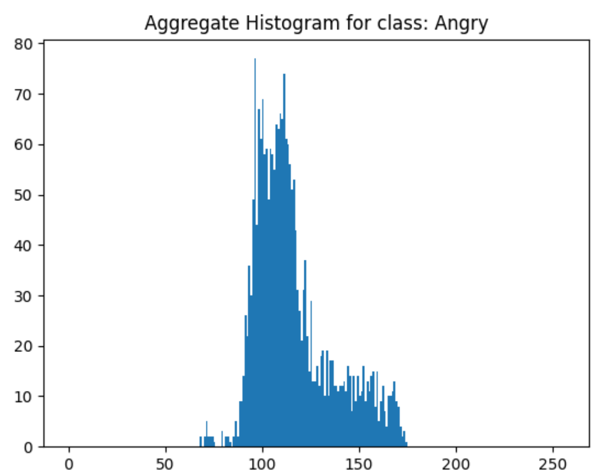
In addition, the way we solved the first two issues was by establishing that a Focused person, on average, should have a slight frown or squint near their glabella if we consider the different angles the A.I. could observe students in a classroom. Thus, this made us realize that the focused label could naturally overlap with many of the neutral and angry images and a few of the happy ones. By that logic, when it came to populating the training and testing datasets for the Focused class, we thought it would be efficient to manually select images from the Neutral, Angry, and Happy classes that could also be labeled as Focused, essentially merging subsets of those three other classes into the Focused one.

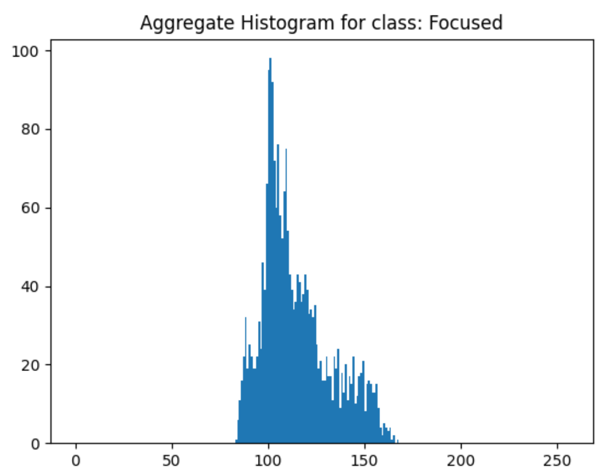
Dataset Visualization

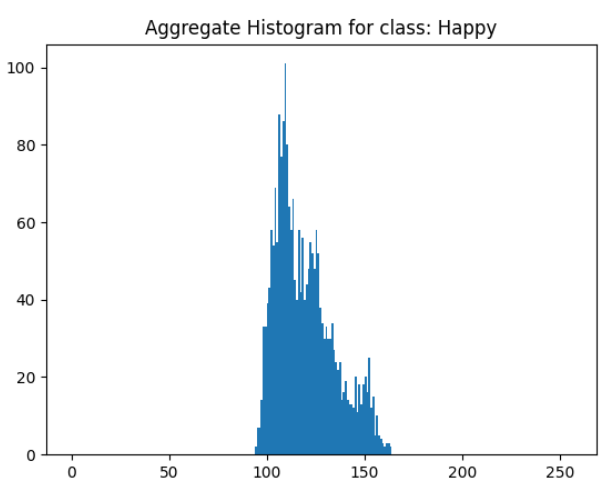
Class Distribution:

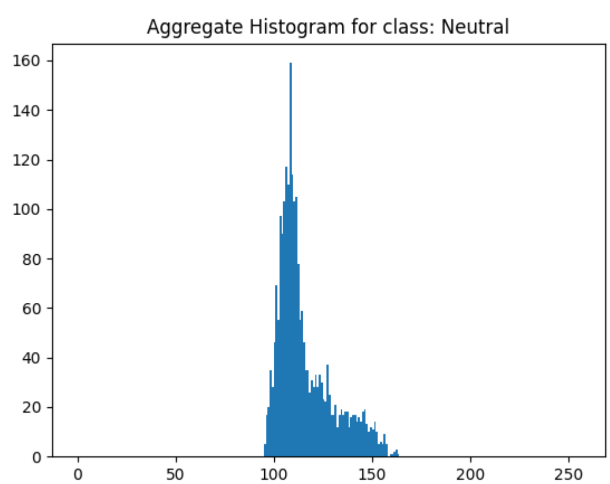


Pixel Intensity Distribution: (Training datasets)



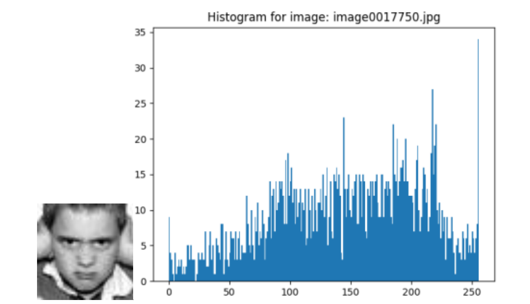


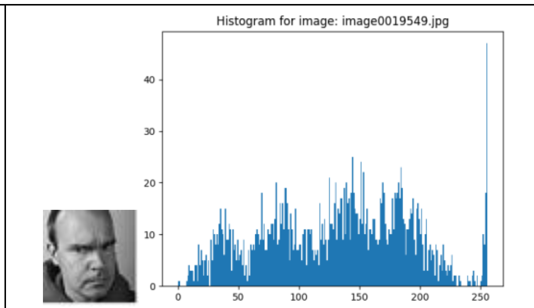


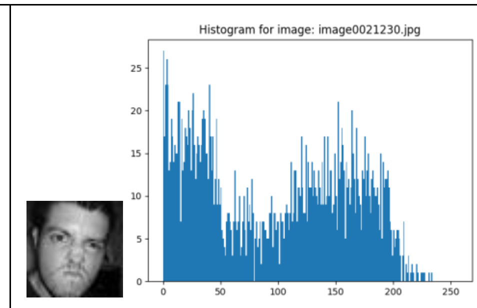


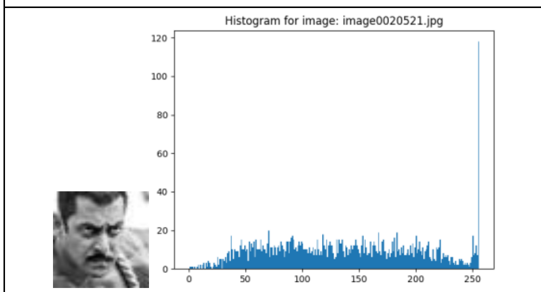
Sample Images:

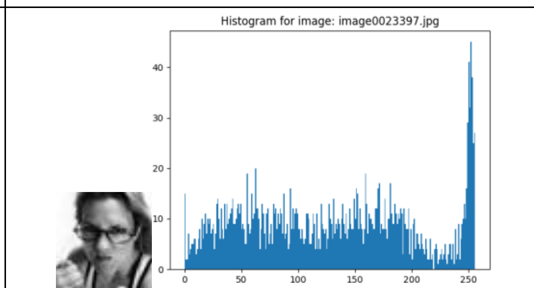
Angry:

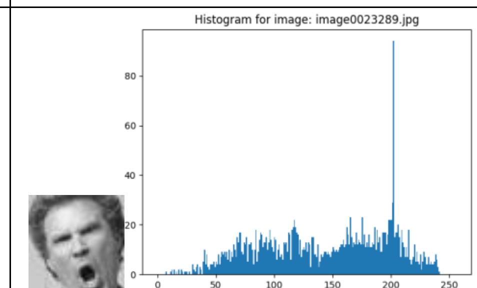


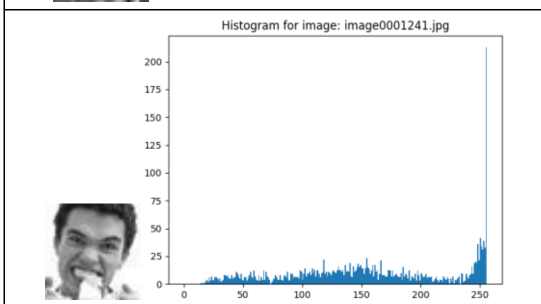


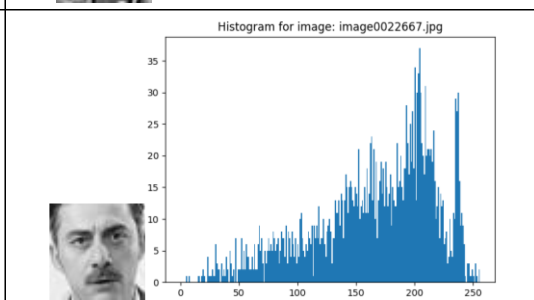


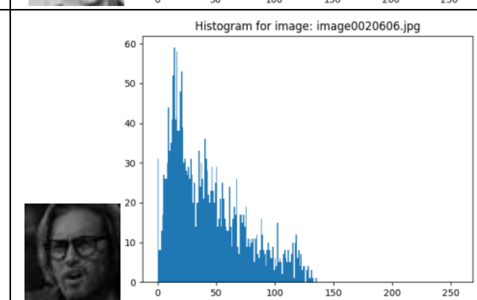


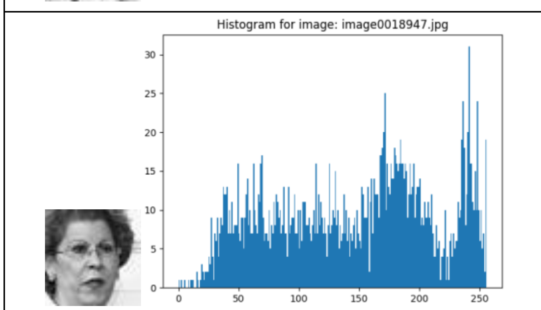


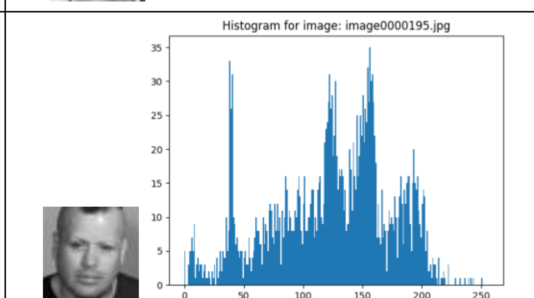


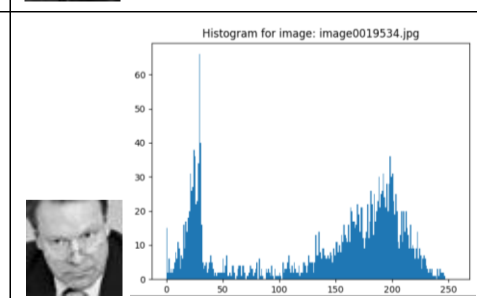


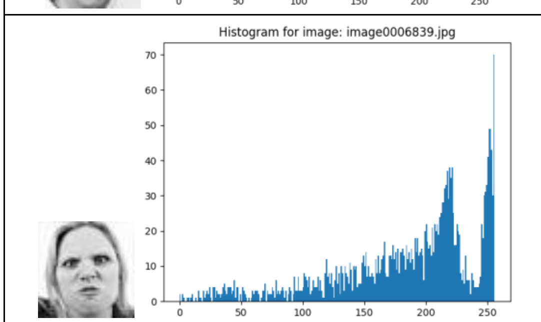


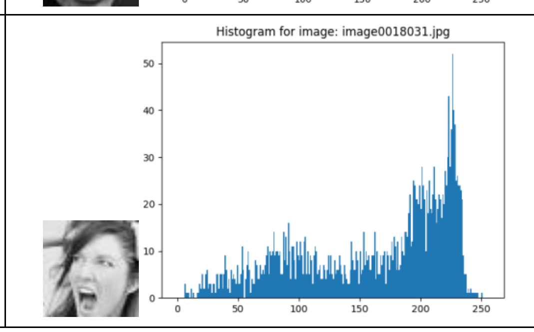


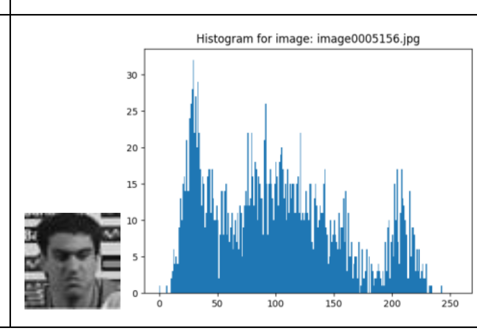




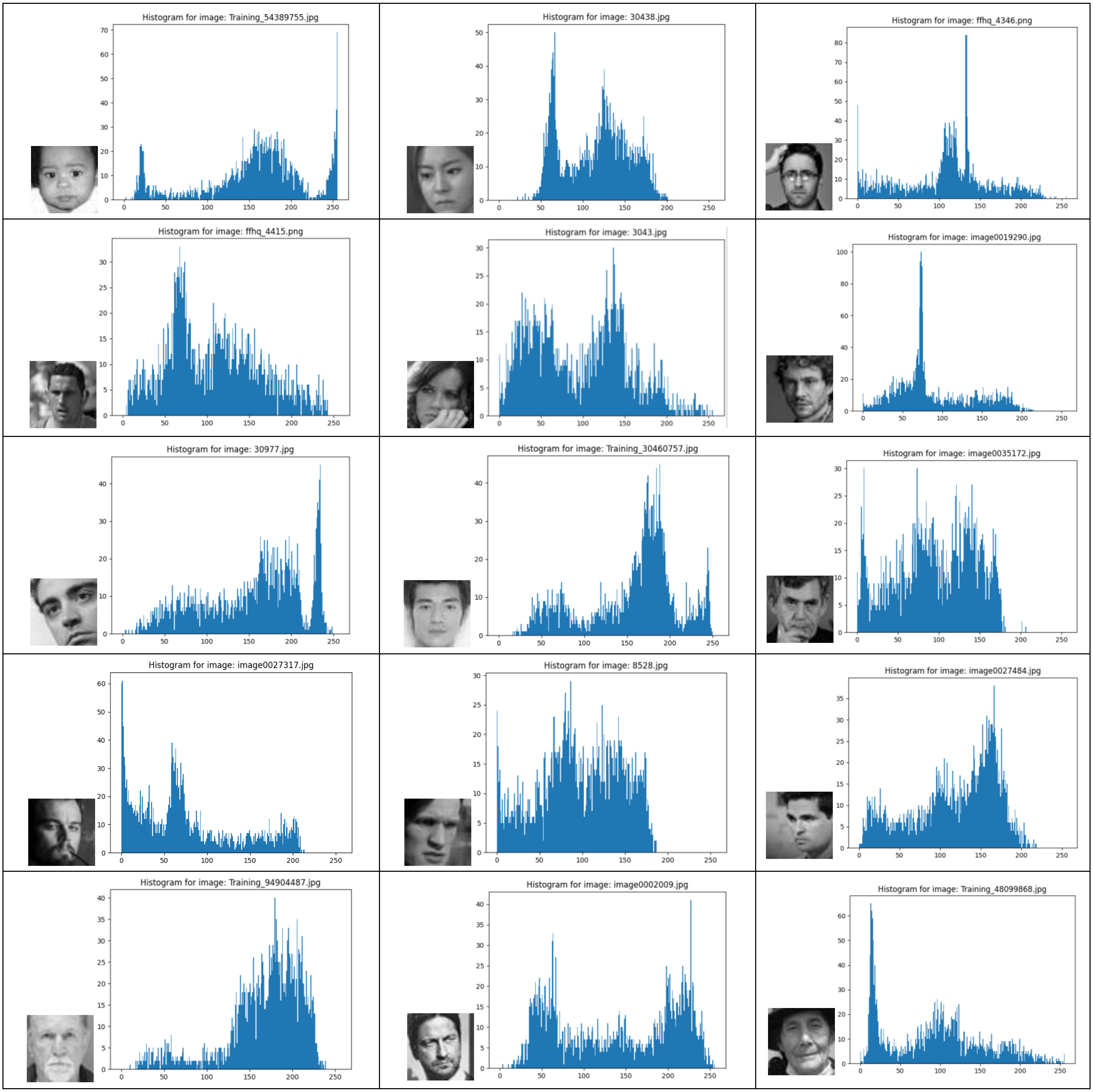




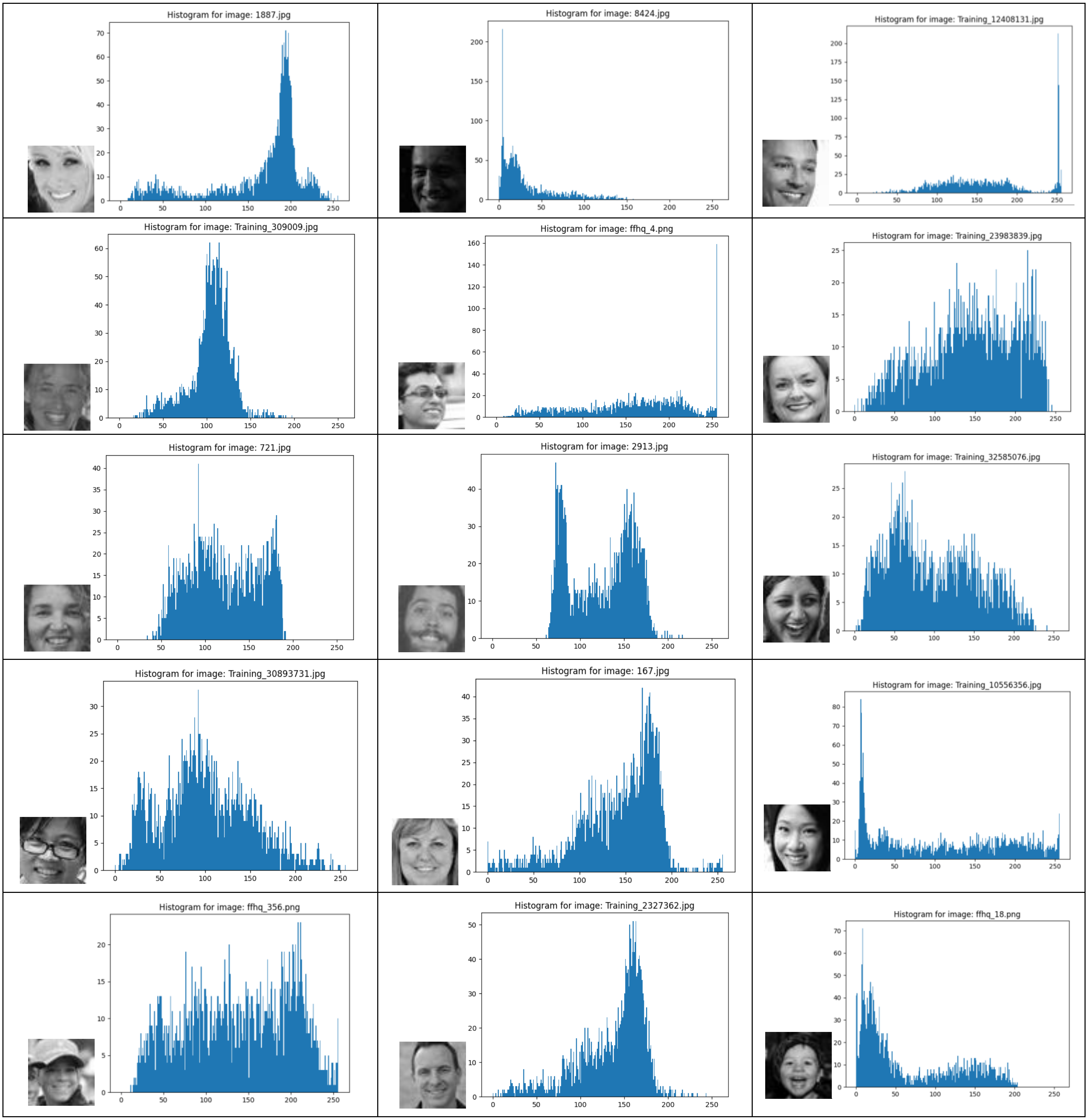




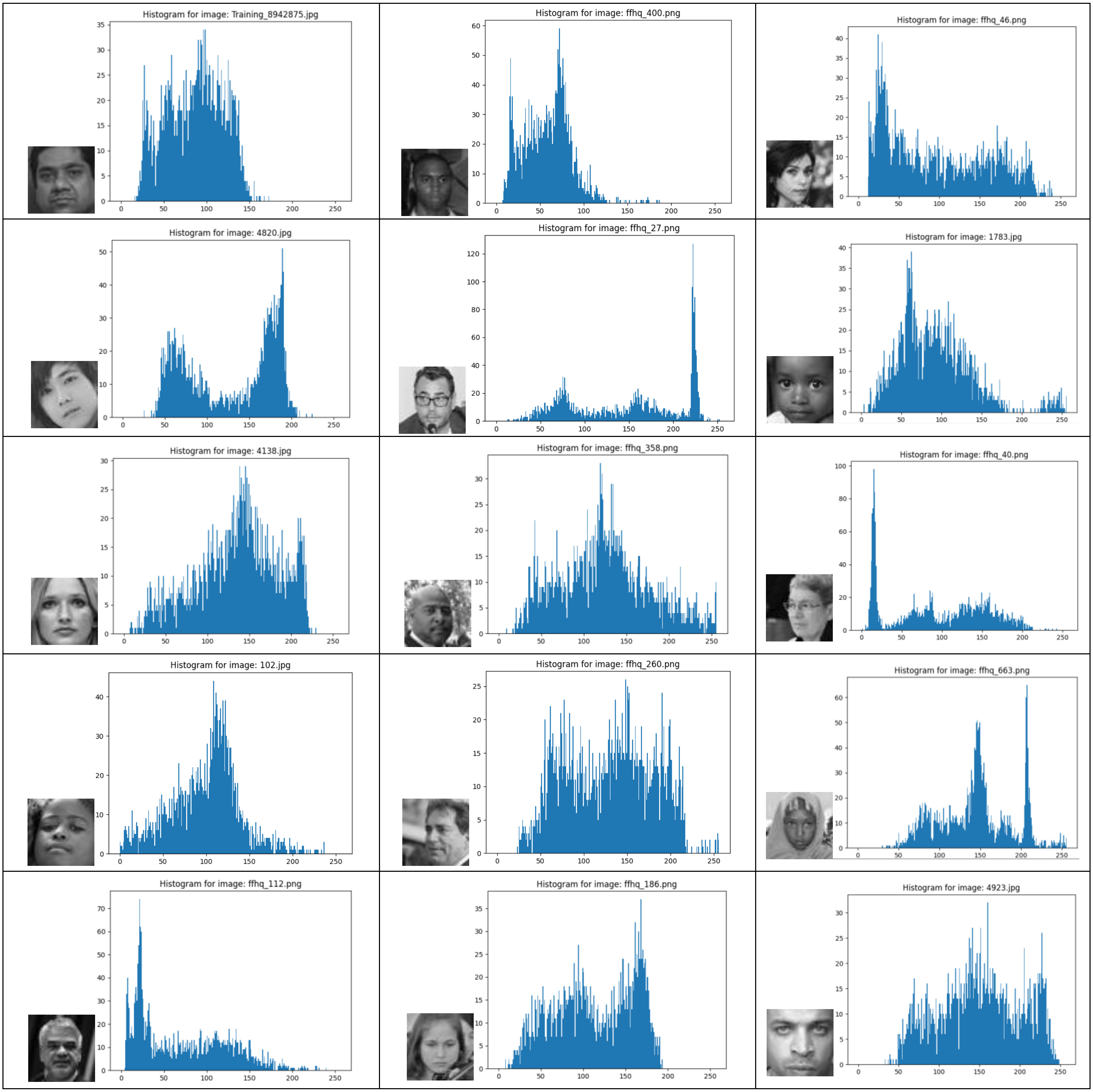
Focused:


Histogram for image: Training_54389755.jpg
Histogram for image: 30438.jpg
Histogram for image: ffhq_4346.png
Histogram for image: ffhq_4415.png
Histogram for image: 3043.jpg
Histogram for image: image0019290.jpg
Histogram for image: 30977.jpg
Histogram for image: Training_30460757.jpg
Histogram for image: image0035172.jpg
Histogram for image: image0027317.jpg
Histogram for image: 8528.jpg
Histogram for image: image0027484.jpg
Histogram for image: Training_94904487.jpg
Histogram for image: image0002009.jpg
Histogram for image: Training_48099868.jpg

Happy:


Histogram for image: 1887.jpg
Histogram for image: 8424.jpg
Histogram for image: Training_12408131.jpg
Histogram for image: Training_309009.jpg
Histogram for image: ffhq_4.png
Histogram for image: Training_23983839.jpg
Histogram for image: 721.jpg
Histogram for image: 2913.jpg
Histogram for image: Training_32585076.jpg
Histogram for image: Training_30893731.jpg
Histogram for image: 167.jpg
Histogram for image: Training_10556356.jpg
Histogram for image: ffhq_356.png
Histogram for image: Training_2327362.jpg
Histogram for image: ffhq_18.png

Neutral:


Histogram for image: Training_8942875.jpg
Histogram for image: ffhq_400.png
Histogram for image: ffhq_46.png
Histogram for image: 4820.jpg
Histogram for image: ffhq_27.png
Histogram for image: 1783.jpg
Histogram for image: 4138.jpg
Histogram for image: ffhq_358.png
Histogram for image: ffhq_40.png
Histogram for image: 102.jpg
Histogram for image: ffhq_260.png
Histogram for image: ffhq_663.png
Histogram for image: ffhq_112.png
Histogram for image: ffhq_186.png
Histogram for image: 4923.jpg

References

[1] N. Segal, “Facial Expressions Training Data.” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/noamsegal/affectnet-training-data

[2] S. Eslamifar, “Facial Emotion Expressions.” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/samaneheslamifar/facial-emotion-expressions

[3] M. Sambare, “FER-2013.” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013

[4] M. bin Mohammed, “How Do I Resize an Image Using PIL and Maintain its Aspect Ratio?” StackOverflow. https://stackoverflow.com/questions/273946/how-do-i-resize-an-image-using-pil-and-maintain-its-aspect-ratio/36036843#36036843 (accessed May 28, 2024).

[5] “Image Module: The Image Class.” Pillow (PIL Fork) 10.4.0.dev0 Documentation. https://pillow.readthedocs.io/en/latest/reference/Image.html?fbclid=IwZXh0bgNhZW0CMTAAAR2XZz_SRzwhHFOFcOi30RXspMTIBU_aeSOM2IMPM6FUK9fTJ3qfpuJeG9w_aem_ARWc0VPTPi0042BbGAiRGrHaOfEm3Vrx0-V7zv8x3_Eb_3Dpzmze_k3tRZNVyqpYK0nqhG4KUyr6QdHRrfhD1c7R#the-image-class (accessed May 28, 2024).

[6] “OpenCV-Python 4.9.0.80.” PyPi. https://pypi.org/project/opencv-python/?fbclid=IwZXh0bgNhZW0CMTAAAR08J1mZAIku5R0zIq-gF_1uczDReNQsYrySI9fef9glL3IGL9fO6VEouAg_aem_AY3Ap_u7uEunF-rLLCjDFydCGsKIKWPbO0Zp1wxm03FG8MmkKlTwSlYj84pJdrPhTMw9Eqw9xxBLW2i2JVDfqSwq#installation-and-usage (accessed May 28, 2024).

[7] “Numpy.ravel.” NumPy. https://numpy.org/doc/stable/reference/generated/numpy.ravel.html?fbclid=IwZXh0bgNhZW0CMTAAAR2lwRunELZ1sjOMrM1-31R4wmWLsjEaVqpFWMzD1mlNk6rRDXDmAAJB7ow_aem_ARVcvQFGFi25HoaP2hYFUG8E1MjOdP3zgvPkaoii7YFFAV6qQgkXe7mdIPgRbORynGthyTFb6u0NbRY8AZ6IFBPQ (accessed May 28, 2024).

[8] “Matplotlib Bars.” W3 Schools. https://www.w3schools.com/python/matplotlib_bars.asp?fbclid=IwZXh0bgNhZW0CMTAAAR2XZz_SRzwhHFOFcOi30RXspMTIBU_aeSOM2IMPM6FUK9fTJ3qfpuJeG9w_aem_ARWc0VPTPi0042BbGAiRGrHaOfEm3Vrx0-V7zv8x3_Eb_3Dpzmze_k3tRZNVyqpYK0nqhG4KUyr6QdHRrfhD1c7R (accessed May 28, 2024).

[9] “matplotlib.pyplot.hist.” Matplotlib. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html?fbclid=IwZXh0bgNhZW0CMTAAAR3jTgWgo7whad-hIja9XGaB52c8Xrvknw-Ygb157lbIH-8t9DQ-

RmPhTcY_aem_ARVQivLmih6zkUQHuiVppeEzByyIb5x0oTjWIZtHSs_E-7Tgk9yIS-kWCvXc12xbbvVaF7M27cqgDwioadL7Ef9s (accessed May 28, 2024).

[10] "matplotlib.pyplot.bar." Matplotlib. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html?fbclid=IwZXh0bgNhZW0CMTAAAR0ZcYnklduTvu9Lx0H4c1dI-YXsi6du82QBIJUrWtbecZGEC6KB6Hxv1qA_aem_ARUPA-drCT86OnxYJI8rivsePJny-srJ0DZwKMF2ETAzzTxNe5T63ZmWI_2hFbRvHMM_Jy1w6-tx-CeIv4a5PHAy (accessed May 28, 2024).