Emotion Based Image Enhancement

Edit the images to increase the intensity of the given emotion by producing visually plausible images

Thomas Théo Nehemie Benchetrit
École polytechnique fédérale de Lausanne
thomas.benchetrit@epfl.ch

Lucas Strauss
École polytechnique fédérale de Lausanne
lucas.strauss@epfl.ch

Philip Ngo
École polytechnique fédérale de Lausanne
philip.ngo@epfl.ch

Abstract

Image enhancement is a well-researched field with many papers written on this topic. In this paper, we investigate and create an emotion-based image enhancement system by using pre-defined methods and algorithms. This system edits the images to increase the intensity of a given emotion to produce plausible images.

1. Introduction

Visual stimuli create emotional responses in the viewer. The response can stem from the content of the image or the visual characteristics such as colors, contrast, and brightness. According to Russell's emotion model, we have selected 4 emotions, namely: happy, sad, fear and calm [1].

Image enhancement is the procedure of improving the quality and information content to produce visually more appealing images. In this study, we introduce a human-centric emotion elements for image enhancement to mimic the professional photo editing process. In this project, we will determine the components that make an image raise a particular emotion. The dataset will be collected from online sources by using the 4 emotion keywords.

1.1. Goals

The goal of this project is to edit images to increase the intensity of the given emotion by producing visually plausible images.

To enhance the images, the pipeline presented in this report will take in the image that has to be modified and it will apply on it several adjustments independently. First it will adjust the colors, then the contrast, then the colorfulness and finally the luminance. All of these adjustments will differ depending on the emotion that was chosen by the user. At the end of the pipeline, the obtained image should evoke the target emotion more than the original image.

2. Feature Extraction

In order to modify specific features of the image we want to enhance, we build a list of "ideal" features for each emotion that we want to target. That means that for each emotion, it is first needed to extract a color palette that represent this emotion, and so on for all the other features that are going to be used to modify an image.

2.1. Color Palette

What we call the color palette is the K most frequent colors of an image extracted using K-means clustering in the RGB color space. In our case, we used $K = 5$ different colors.

To get a color palette of a large data set of images, we build a large matrix that contains all the color palettes of all the images in the data set. With this matrix, we do a final color extraction on the matrix and get the colors that return the most frequent colors out of all images. The process is demonstrated in Figure 1.

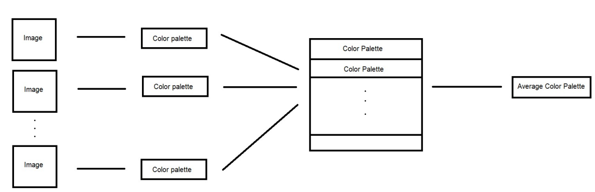

Figure 1. Average image pipeline

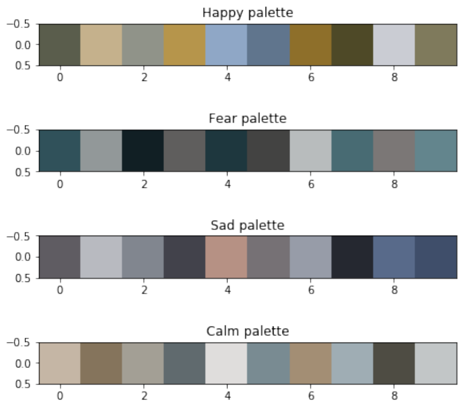

Figure 2. The color palette for each emotion of our dataset

2.2. Contrast

We use the following formula to define contrast in the LAB color space[5]:

$$\frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

The minimum and maximum are computed by eroding and dilating, respectively, the image in 5×5 regions. Then, we average the above formula over the entire image and get the percentage of contrast.

2.3. Colorfulness

Using the formula found in [2], we compute the percentage of colorfulness of an image with equation below where R, G, B are the red-, green-, and blue channels and σ, μ are the standard deviation and mean respectively.

$$\begin{aligned} rg = R - G, yb = \frac{1}{2}(R+G) - B \\ \sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \\ \mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \\ C = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb} \end{aligned}$$

2.4. Luminance

We define luminance as the mean brightness of the image. To compute it we simply convert the RGB image to the HSV color space and compute the average of the $value$ channel.

3. Transfer

3.1. Color

To implement the color transfer, we based it off on the paper ”Color Transfer between Images” by Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley from the University of Utah [4]. We start by converting the RGB images into the CIELAB colorspace. Then we compute the mean and standard deviation of the target image and the source image and get l'_s, a'_s, b'_s, which is the lab-axis of the final image after the color transformation, according to the equation below. The equation below denotes σ_t^l, σ_s^l, l_s, $\tilde{l}_s$, $\tilde{l}_t$ as standard deviation for target image l-axis, standard deviation for source image l-axis, source image l-axis, mean for source image l-axis and, mean for target image l-axis respectively. The a- and b-axis follow the same structure.

$$\begin{aligned} l'_s &= \frac{\sigma_t^l}{\sigma_s^l} \cdot (l_s - \tilde{l}_s) + \tilde{l}_t \\ a'_s &= \frac{\sigma_t^a}{\sigma_s^a} \cdot (a_s - \tilde{a}_s) + \tilde{a}_t \\ b'_s &= \frac{\sigma_t^b}{\sigma_s^b} \cdot (b_s - \tilde{b}_s) + \tilde{b}_t \end{aligned}$$

The image is then converted back into RGB color space. In this case, the target image is the color palette obtained from the color extraction mentioned in section 2.1. It is also mentioned in the paper that the result's quality depends on the images' similarity in composition but when used for this project we only use the color palette which yields good results for some and less for others. We have not found a direct correlation of what that could be.

What differs from the paper is that after this process, we then choose a random sample image from our set of images that match a certain emotion. We then set that as our target image and run it through the same procedure to create another color-transferred image. We finally use our original source image, the two images obtained, and overlay them and decrease the opacity of each image to $\frac{1}{3}$ which is illustrated in Figure 3. This gives the images a more natural look and gives variety in the final image produced by the pipeline.

3.2. Contrast

To adjust the contrast of an image, we simply weight the RGB image by `target_contrast / initial_contrast` to scale every pixel at the target value.

3.3. Colorfulness

Adjusting the colorfulness of an HSV image is as simple as multiplying the saturation channel by a factor. In our case, we want the image to have a specified colorfulness feature, so the factor is given by `target_colorfulness / initial_colorfulness`.

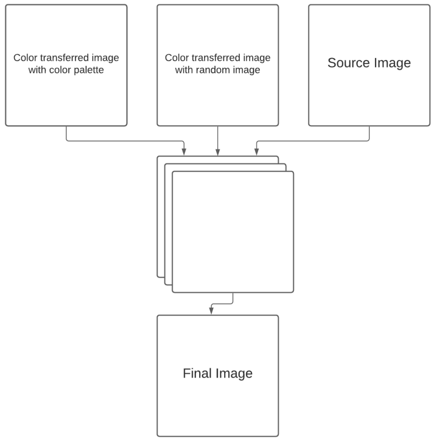

Figure 3. Color Transfer

3.4. Luminance

Given the target luminance t, the actual luminance value of the image l and the rate of change r, we adjust the value of the image in the HSV color space according to $v' = v \cdot (t - v) * r$.

4. Pipeline

In our pipeline, at each step, we create two images. The first one using the average feature of the emotion, and the second one using the feature of the randomly chosen image.

Our feature transfer pipeline for one image and one emotion operates as follows:

1. We first compute and average the value of each feature for this emotion (color palette, contrast, colorfulness, and luminance).

2. We randomly chose an image in the set of the target emotion images and extract all the features as well.

3. We create two images: the first one, a_1, by transferring the average emotion color palette, and the second one, r_1, by transferring the randomly chosen image color palette. We then combine them with the original image o using the weights $o_1 = \frac{1}{8}o + \frac{1}{3}a_1 + \frac{1}{3}r_1$.

4. We adjust the contrast of the image obtained at the previous step, o_1, by transferring the average contrast of the emotion to a_2, and the contrast of the randomly chosen image to r_2. We then generate $o_2 = \frac{3}{4}o + \frac{1}{8}a_2 + \frac{1}{8}r_2$. Because the contrast feature is very sensitive and varies a lot between images, adjusting only $\frac{1}{4}$ of it changes enough in the image to influence the conveyed emotion. The weights were chosen experimentally after trying several different combinations.

5. We then adjust the colorfulness and the luminance similarly: by transferring the average feature and the feature of the randomly chosen image separately and then combining the images using weights of $\frac{1}{3}$ for each one.

We picked 250 neutral images and ran them through our pipeline, generating 1000 images in total (one image per emotion).

5. Experiment

5.1. Image research

First, we gathered images corresponding to the different emotion keywords from the Instagram platform using a scrapper script. By searching for images with some specific hashtags like *#landscape, #panorama*, and by targeting some specific photographer accounts, we were able to assemble 250 images for each emotion and 250 neutral images that we used to test our method.

Gathering images in a fully automated way was not possible because we needed the images to make sure images were not containing any faces and that they were really corresponding to the specific emotion we needed. So some manual work was done to validate the dataset and filter out inaccurate pictures.

5.2. Amazon Mechanical Turk

We compare a state-of-the-art image enhancement algorithm [3] with our more classical approach of image manipulation. Using Amazon's Mechanical Turk platform, we conducted an experiment to determine which one between our pipeline or the *UEGAN* is the best at enhancing a particular emotional response for the viewer.

We randomly displayed to the participants a targeted emotion, and the original, the GAN modified and our pipeline-modified images according to the displayed emotion, and the participants had to rank the three images according to which image best represent the targeted emotion. Then, when the ranking is obtained, the first image get 3 points, the second get 2 point and the last get one point.For each type of image (pipeline, gan, original), the score is computed as the number of points of a model over the total number of points.

6. Results

Our proposed pipeline produces 4 versions of the input image, one for each happy, sad, fear and calm emotion.

From the experiment described, we can export the result of our pipeline over the four emotions. All the results are presented on Figure.6,7,8,9

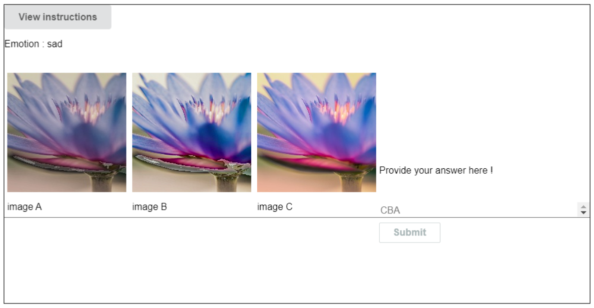

Figure 4. Example of the Amazon's Mechanical Turk Experiment

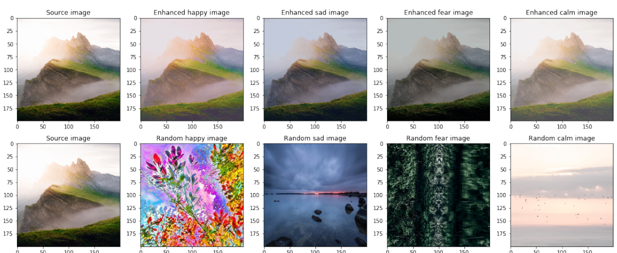

Figure 5. Example of what our pipeline produces

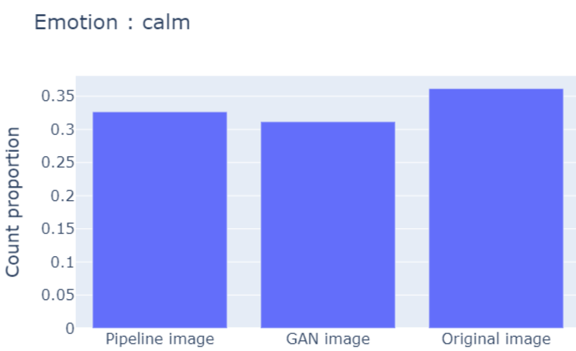

Figure 6. Result of the experiment for the "calm" emotion

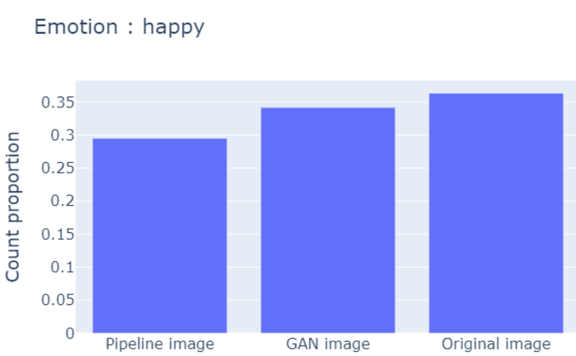

Figure 7. Result of the experiment for the "happy" emotions

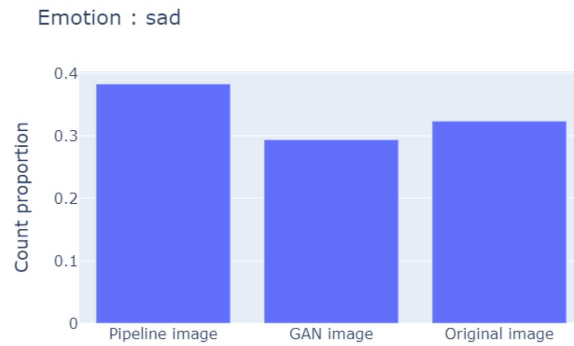

Figure 8. Result of the experiment for the "sad" emotion

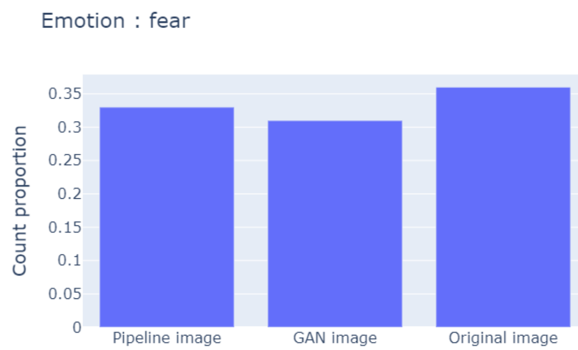

Figure 9. Result of the experiment for the "fear" emotion

7. Discussion

From the Figure.6,7,8,9, the first thing that is worth noting is that for almost all the emotions, the original image yield better results than the modified images, either with the GAN or our Pipeline. This may be because, the workers that are available in the mechanical turk may not be reliable and prefer to choose the most visually pleasing image over the one that best represent the emotion. As the original image is not modified and is already edited professionally, it will often be more pleasing to the eye. However, when taking into account only the modified images, our model yields better result than the GAN, except for the Happy emotion.

One surprising result of the experiment is that our model does not perform the same for each emotion. Indeed, for sadness, our pipeline returns much better performance than the other two images. This may come from several reasons. First it may be possible that the dataset created for the sad images may be more accurate. Another explaination may be found in the fact that when computing the color palette for each emotion, it usually returns a color palette which has darker tones. Therefore, as usually sadness is displayed through film colorimetry as washed colors and darker tones, that may be why our pipeline yielded its best result with the ”sad” emotion.

8. Conclusion

In conclusion, it is possible to achieve meaningful results with a fixed set of algorithms and produce plausible images. Our model performed very well for some images, and poorly for others. For the experiment with Amazon’s Mechanical Turk, we noticed that our model performance depend on the emotion chosen and is the most efficient for the ”sad” emotion. There is still a lot of room for improvement, and the biggest one is to identify why some images work better than others, and try to counteract that. Another can be to ask more specific question for to the user or test on a larger group of people.

Finally, finding images that convey a certain emotion will never be fully one-sided since depending on the culture, the same colors may represent different emotion. Therefore it could be a congruent experiment sequel to train our model based on specific cultural pool of images.

References

[1] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. 11 2018.

[2] D. Hasler and S. E. Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, volume 5007, pages 87–95. International Society for Optics and Photonics, 2003.

[3] Z. Ni, W. Yang, S. Wang, L. Ma, and S. Kwong. Towards unsupervised deep image enhancement with generative adversarial network. *IEEE Transactions on Image Processing*, 29:9140–9151, 2020.

[4] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21:34–41, 10 2001.

[5] Wikipedia. Contrast (vision) — Wikipedia, the free encyclopedia, 2021.