Name: **Iman Mousaei** Student Number: **98222099**
Course: **Image Processing** Date: April 22, 2023
Assignment **1**

1 Exercise 1

We can use OpenCV library and its *detectMultiScale* function to detect faces. But how does it work?

The *detectMultiScale* function from OpenCV is used for object detection in images or videos. It uses a technique called Haar cascades or Cascade classifiers for object detection. The training of a cascade classifier typically involves the following steps:

1. Positive and negative examples: The first step is to collect a large number of positive and negative examples of the object to be detected.

2. Feature extraction: Next, a set of features is extracted from the positive and negative examples. The features used in cascade classifiers are typically Haar-like features, which are rectangular features that can be computed very efficiently using the Integral Image.

3. AdaBoost training: AdaBoost is a machine learning algorithm that is used to train a classifier for distinguishing between positive and negative examples of the object. During the training process, AdaBoost selects the best features and trains a weak classifier for each feature. The weak classifiers are combined to form a strong classifier.

4. False positive reduction: After the AdaBoost training, the strong classifier is applied to the training set to detect the positive examples. However, the classifier may also detect false positives in the negative examples. To reduce the false positive rate, a technique called bootstrapping is used, where the false positive examples are added to the negative set, and the classifier is retrained.

5. Cascade structure: Once the false positive rate is reduced to an acceptable level, the strong classifier is organized into a cascade structure. The cascade consists of multiple stages, where each stage consists of a set of weak classifiers. The output of each stage is used to determine whether to continue to the next stage or not. If a region of the image is classified as negative at any stage, it is rejected as a potential object, and the detection process stops for that region.

And now we perform some geo transformations on the faces. Here are some examples:
Flip:

Rotation:

Shift:

2 Exercise 2

2.1 a

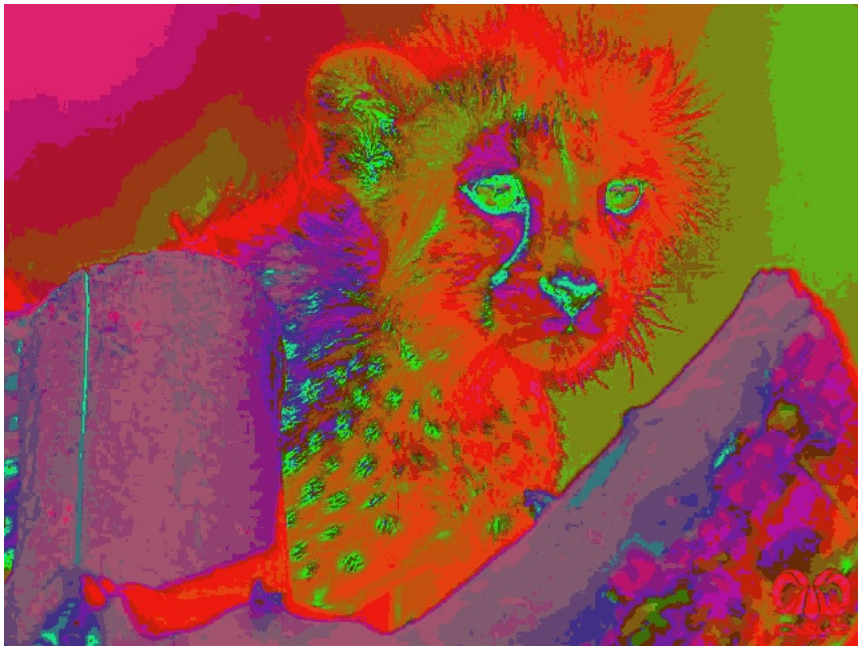
For compressing an image using indexing(with k colors), we first find the k colors with the highest frequency. Then we utilize k-means to cluster the colors into k clusters with the centers of the most common colors. Then, we replace each color with the color of its cluster center. This way we only use k colors, therefore, resulting in less size. We can see the result of $k = 8$ here:

2.2 b

We learned HSV, YCrCb and LAB. We can use *cvtColor* function to convert the color representations to each other. You can see the results here: TODO Here we rendered LAB as BGR and compressed it using indexing of $k = 25$:

And here we did the same to YCrCb with $k = 25$:

And now for HSV and $k = 25$:



References

[1] https://scikit-learn.org/

[2] https://stackoverflow.com/

[3] https://towardsdatascience.com/