A Simple Application: Image Compression Using the K-means Clustering Algorithm

Xi XIA
International School
Beijing University of Post and Telecommuniccation
Beijing, China
xiaxi-sybil@bupt.edu.cn

***Abstract*— This paper uses the Lloyd's K-means Clustering algorithm with RGB tuples in order to find the k colors that best represent the input image. In addition, this program also produces scatter plots of the colors to better visualize the clusters.**

***Keywords*— Image Compressing, K-means**

I. INTRODUCTION

A. *Image compression*

Image Compression is to reduce the amount of data needed to represent digital images.

Image data can be compressed because of the redundancy in the data. The redundancy of image data mainly manifests as: spatial redundancy caused by correlation between adjacent pixels in the image; time redundancy caused by correlation between different frames in the image sequence; correlation of different color planes or spectrum bands Spectrum redundancy. The purpose of data compression is to reduce the number of bits required to represent the data by removing these data redundancy. Since the amount of image data is huge, it is very difficult to store, transfer, and process, so the compression of image data is very important.

The information age has brought about an "information explosion", which has greatly increased the amount of data. Therefore, data transmission needs to be effectively compressed regardless of transmission or storage. In remote sensing technology, various space probes use compression coding technology to send huge information back to the ground.

Image compression is the application of data compression technology on digital images. Its purpose is to reduce redundant information in image data and store and transmit data in a more efficient format.

B. *Basic methods of image compression*

Image compression can be either lossy data compression or lossless data compression. Lossless compression is preferred for technical drawings, diagrams, or comics as drawn, because lossy compression methods, especially at low bit rates, can introduce compression distortion. The compression of such valuable content, such as medical images or scanned images for archiving, also tries to select a lossless compression method. The lossy method is well suited for natural images. For example, in some applications, small loss of image is acceptable (sometimes undetectable), which can greatly reduce the bit rate.

1) *Lossless image compression techniques are:*
- Run-length encoding – used in PCX[1], BMP, TGA, TIFF [2]
- Area image compression
- DPCM and Predictive Coding [3]
- Entropy encoding
- LZW coding
- Huffman coding [4]
- Adaptive dictionary algorithms – used in GIF and TIFF [5]
- Deflation – used in PNG, MNG, and TIFF
- Chain codes

2) *Lossy compression techniques are:*
- Reducing the color space.
- Chroma subsampling. [6]
- Fractal compression [7]
- Transform coding (DCT and Wavelet):
- Discrete cosine transform (DCT) used in JPEG. Before applying DCT, colors are converted to Y′ CBCR, consisting of one luma component (Y'), and two chroma components (CBCR). [8]
- Wavelet transform (reversible or irreversible) used in JPEG2000. [9]

3) *Other compression methods are:*

Resource Interchange File Format (RIFF) used in WebP, which performs image optimization for lossy images with transparency. WebP is based on VP8's intra-frame coding. [10] [11]

C. *Introducing k-Means*

k-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c}\sum_{j=1}^{c_i}(\| x_i - v_j \|)^2 \qquad (1)$$

where,

$\| x_i - v_j \|$ is the he Euclidean distance between xi and vj.

c_i is the number of data points in ith cluster.

c is the number of cluster centers.

1) Algorithmic steps for k-means clustering

Let $X = \{x1, x2, x3, \dots\dots.., xn\}$ be the set of data points and $V = \{v1, v2, \dots\dots., vc\}$ be the set of centers.

a. Randomly select ‘c’ cluster centers.

b. Calculate the distance between each data point and cluster centers.

c. Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

d. Recalculate the new cluster center using:

$$v_i = \left(\frac{1}{c_i}\right)\sum_{j=1}^{c_i} x_i \qquad (2)$$

where,

c_i represents the number of data points in ith cluster.

e. Recalculate the distance between each data point and new obtained cluster centers.

f. If no data point was reassigned then stop, otherwise repeat from step 3).

The algorithm is shown as below:

Algorithm 1 K-means

```
randomly choose k examples as initial centroids
**while true:**
    create k clusters by assigning each example to closest
    centroid
    compute k new centroids by averaging eamples in each
    cluster
    **if** (centroids don’t change)
        break
    **end if**
**end while**
```

Algorithm 2 Choose The Number Of Clusters k

```
best = kMeans(points)
**for** t in range numTrials):
    C= kmeans(ponints)
    **if** ( dissimilarity(C) < dissimilarity(best) )
        best = C
    **end if**
**end for**
return best
```

2) Advantages

a. Fast, robust and easier to understand.

b. Relatively efficient: $O(tknd)$, where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, $k, t, d \ll n$.

c. Gives best result when data set are distinct or well separated from each other.

3) Disadvantages

a. The learning algorithm requires a priori specification of the number of cluster centers.

b. The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.

c. The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get.

d. different results (data represented in form of cartesian co-ordinates and polar co-ordinates will give different results).

e. Euclidean distance measures can unequally weight underlying factors.

f. The learning algorithm provides the local optima of the squared error function.

g. Randomly choosing of the cluster center cannot lead us to the fruitful result. Pl. refer Fig.

h. Applicable only when mean is defined i.e. fails for categorical data.

i. Unable to handle noisy data and outliers.

j. Algorithm fails for non-linear data set.

II. Problem Formulation

In our problem of image compression, K-means clustering will group similar colors together into ‘k’ clusters of different colors (RGB values). Therefore, each cluster centroid is the representative of the three-dimensional color vector in RGB color space of its respective cluster. Now, these ‘k’ cluster centroids will replace all the color vectors in their respective clusters. Thus, we need to only store the label for each pixel which tells the cluster to which this pixel belongs. Additionally, we keep the record of color vectors of each cluster center. Look at the original and compressed images is shown in section IV.

III. Implementation

First of all, it’s always good to remember an image is just a vector of pixels. Each pixel is a tuple of three integer values between 0 and 255 (an unsigned byte), which represent that pixel’s color’s RGB values.

We want to use K Means clustering to find the k colors that best characterize an image. That just means we could treat each pixel as a single data point (in 3-dimensional space), and cluster them.

Algorithm 3 k-means Image Compression

```
read image as RGB floats
**for** i in range(num_iterations)
    **for** rgb in enumerate (image_vectors)
        Find the Closest Label via L2 Norm
        Optimize Cluster Prototypes (Center of Mass of
        Cluster)
    **end for**
    **for** k_i in range(k-means)
        Find Current Distortion Distances
        end for
    end for
    return labels, clusters prototypes
```

IV. Results

The following results are the outcomes after 20 iterations.

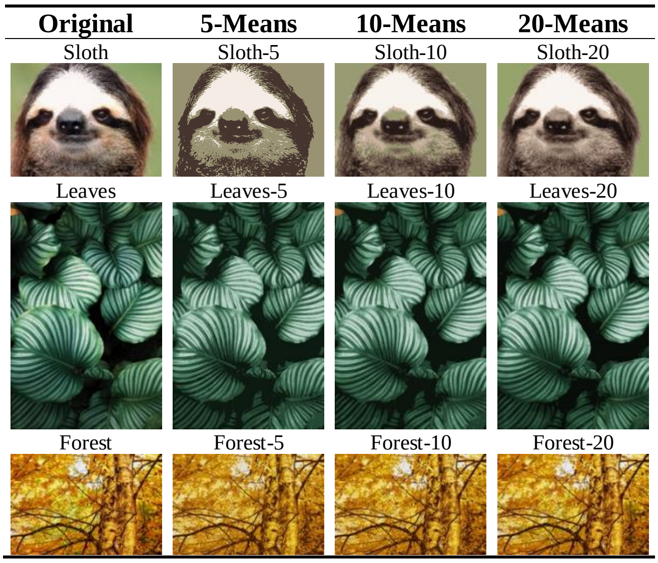


Figure1. K-means Image Compressing Results - 1

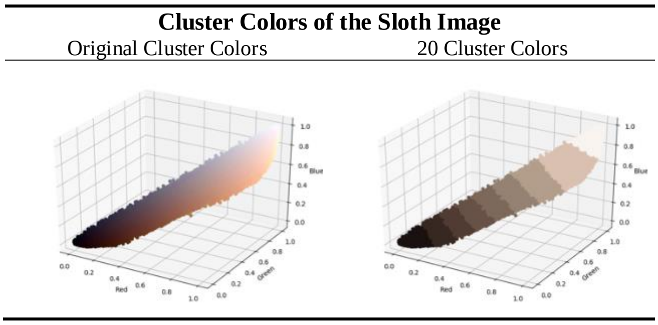


Figure 2. Cluster Colors of the Sloth Image

As the k is increasing, the image is represented by more colors, and the details are able to be presented.(see Fig.1) At the first row, we can see the effect of compressing is quite like filters. The 5-means sloth looks like the cartoon character in "Crazy Animal City". For the row 2(leaves), there is a small area of yellow leaves. However, it is totally ignored in the 5-means scenario. And in row 3, there is a huge area of light green. Again, it is ignored in 5-means and 10-means scenario. The gradual changes at edges or in the shadow disappear.

So, the area of color and the contrast ratio are both the influence factor. More experiments are needed to define the best k.

By k-means compressing, we are able to show an image using less colors.

Storage Size of Images

	Original	5-Means	10-Means	20-Means
Sloth	42 KB	21KB	47 KB	79 KB
Leaves	334 KB	198 KB	236 KB	342 KB
Forest	3.3 MB	1.6 MB	2.7 MB	4.3 MB

Table 1. K-means Image Compressing Results – 2

As we can see in table.2, the storage size of images became smaller after compressing (k = 5). However, when k is large (k = 20), the image may occupy lager storage space. So, k-means compressing can't guarantee storage size reducing even the image has less color. More experiments are needed to figure out the reason.

V. Conclusion

Using k-means clustering algorithm, we are able to make new images with only 10% of the original's colors, which looked very similar to them. We also got some cool looking filters thanks to K means clustering.

References

[1] V Shantagiri, P. and K N, S. (2013). Pixel Size Reduction Loss-Less Image Compression Algorithm. International Journal of Computer Science and Information Technology, 5(2), pp.87-95.

[2] R, P. and R.J, I. (2013). An Overview of Digital Image Steganography. International Journal of Computer Science & Engineering Survey, 4(1), pp.23-31.

[3] Taheri, A. and Mahdavi-Nasab, H. (2018). Sparse representation based facial image compression via multiple dictionaries and separated ROI. Multimedia Tools and Applications, 77(23), pp.31095-31114.

[4] Hussain, A., Al-Fayadh, A. and Radi, N. (2018). Image compression techniques: A survey in lossless and lossy algorithms. *Neurocomputing*, 300, pp.44-69.

[5] Hussain, A., Al-Fayadh, A. and Radi, N. (2018). Image compression techniques: A survey in lossless and lossy algorithms. Neurocomputing, 300, pp.44-69.

[6] Luo, H., Ci, S., Wu, D. and Tang, H. (2010). End-to-end optimized TCP-friendly rate control for real-time video streaming over wireless multi-hop networks. *Journal of Visual Communication and Image Representation*, 21(2), pp.98-106.

[7] Fisher, Y. (1996). Fractal image compression. New York: Springer.

[8] Rao, K. and Yip, P. (2014). Discrete Cosine Transform. Saint Louis: Elsevier Science.

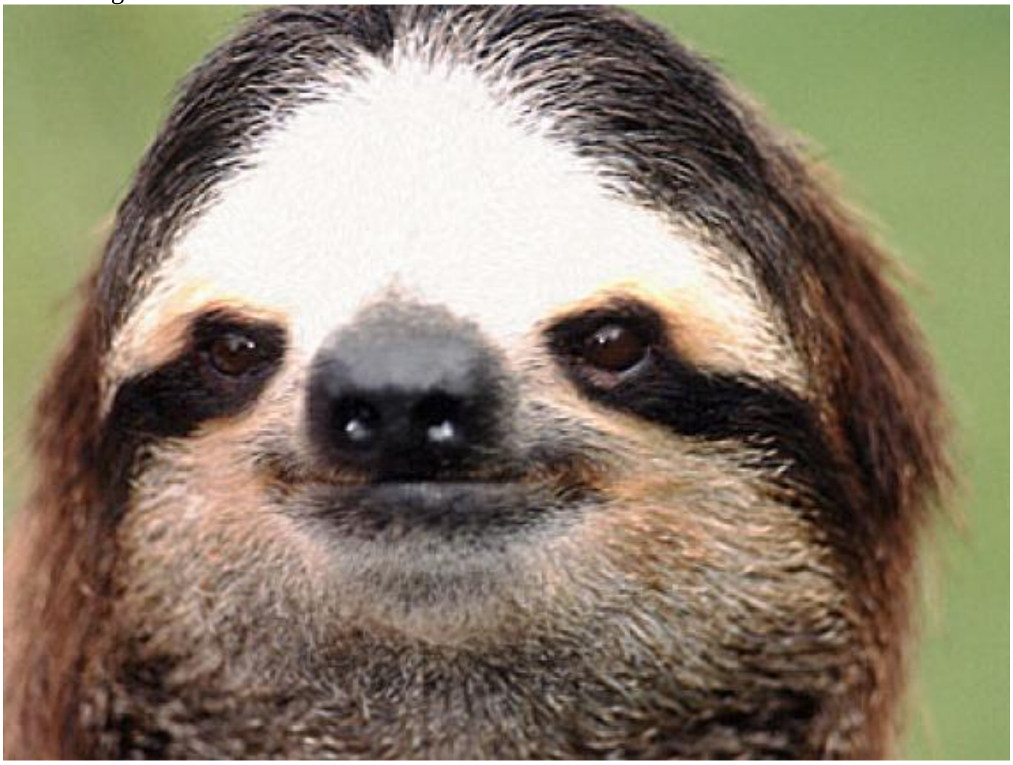
[9] Pan, H., Siu, W. and Law, N. (2007). Lossless image compression using binary wavelet transform. IET Image Processing, 1(4), p.353.

[10] Armasoiu, G. (2015). Manufacturing Specific Information Recapture from Geometric Model Using a Data Interchange Format File. Applied Mechanics and Materials, 809-810, pp.829-834.

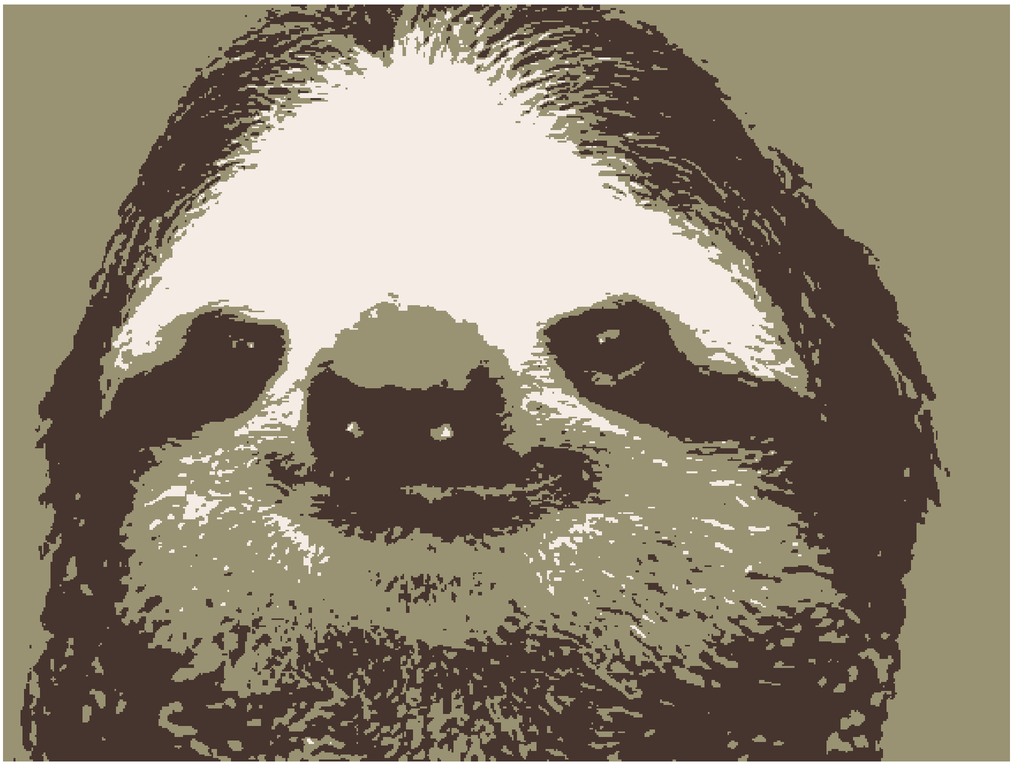
[11] LOW, Y. and BESAR, R. (2004). WAVELET-BASED MEDICAL IMAGE COMPRESSION USING EZW: OBJECTIVE AND SUBJECTIVE EVALUATIONS. Journal of Mechanics in Medicine and Biology, 04(01), pp.93-110.

Appendix – Results

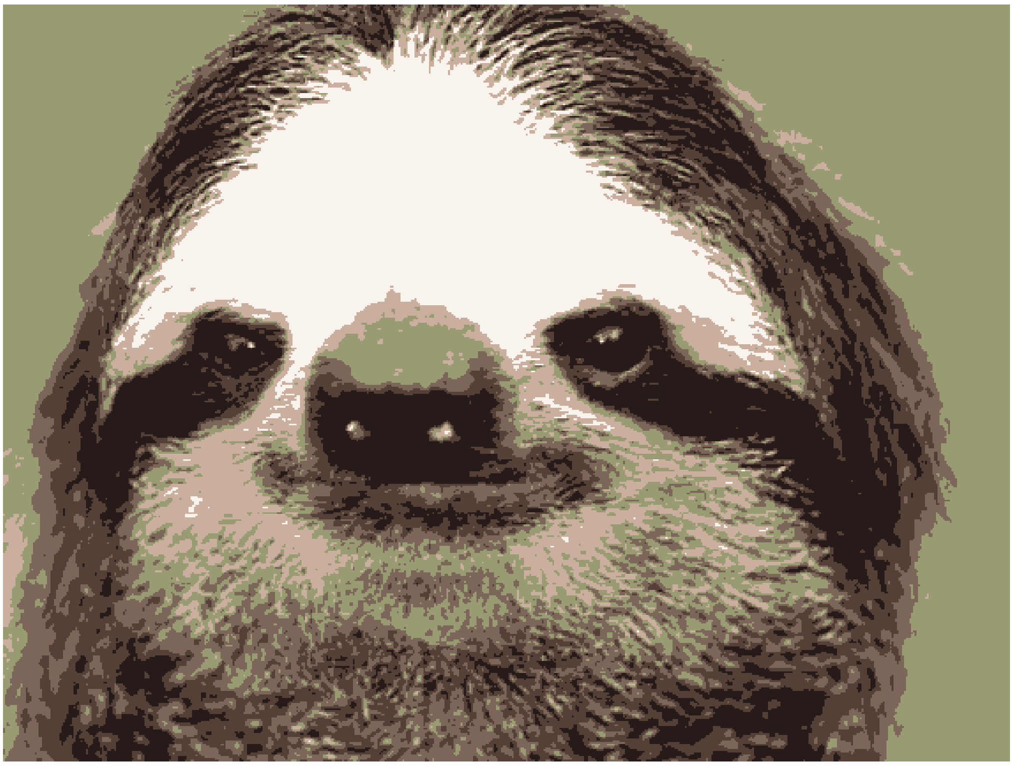
sloth – origin



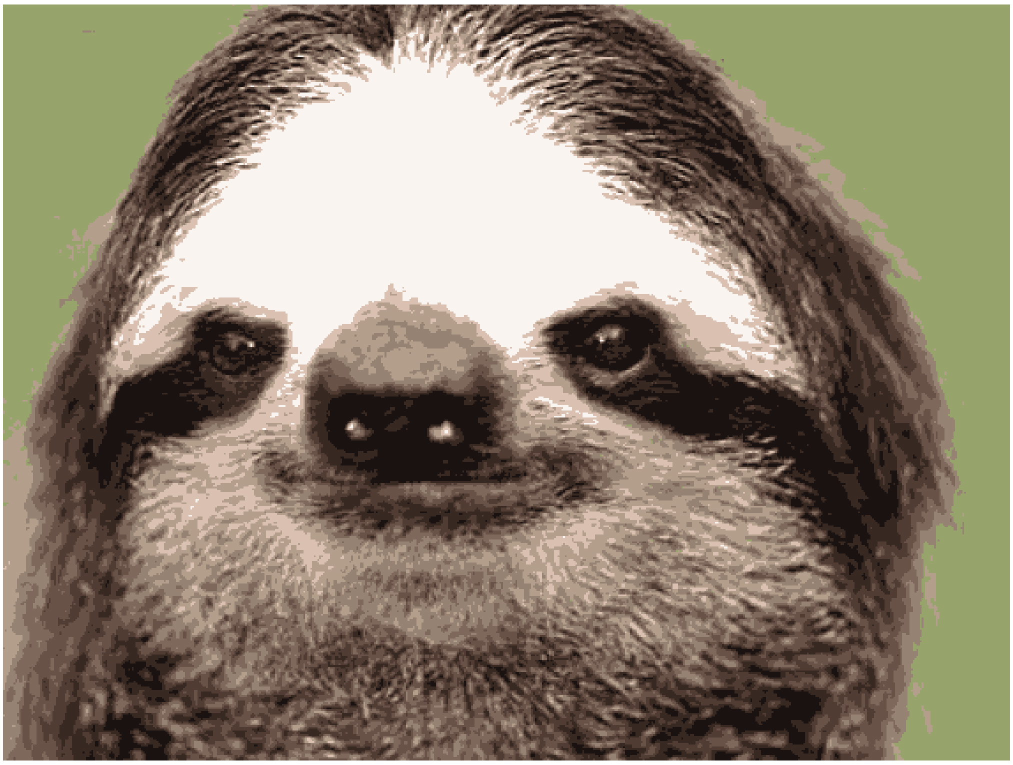
sloth – 5-means



sloth – 10-means

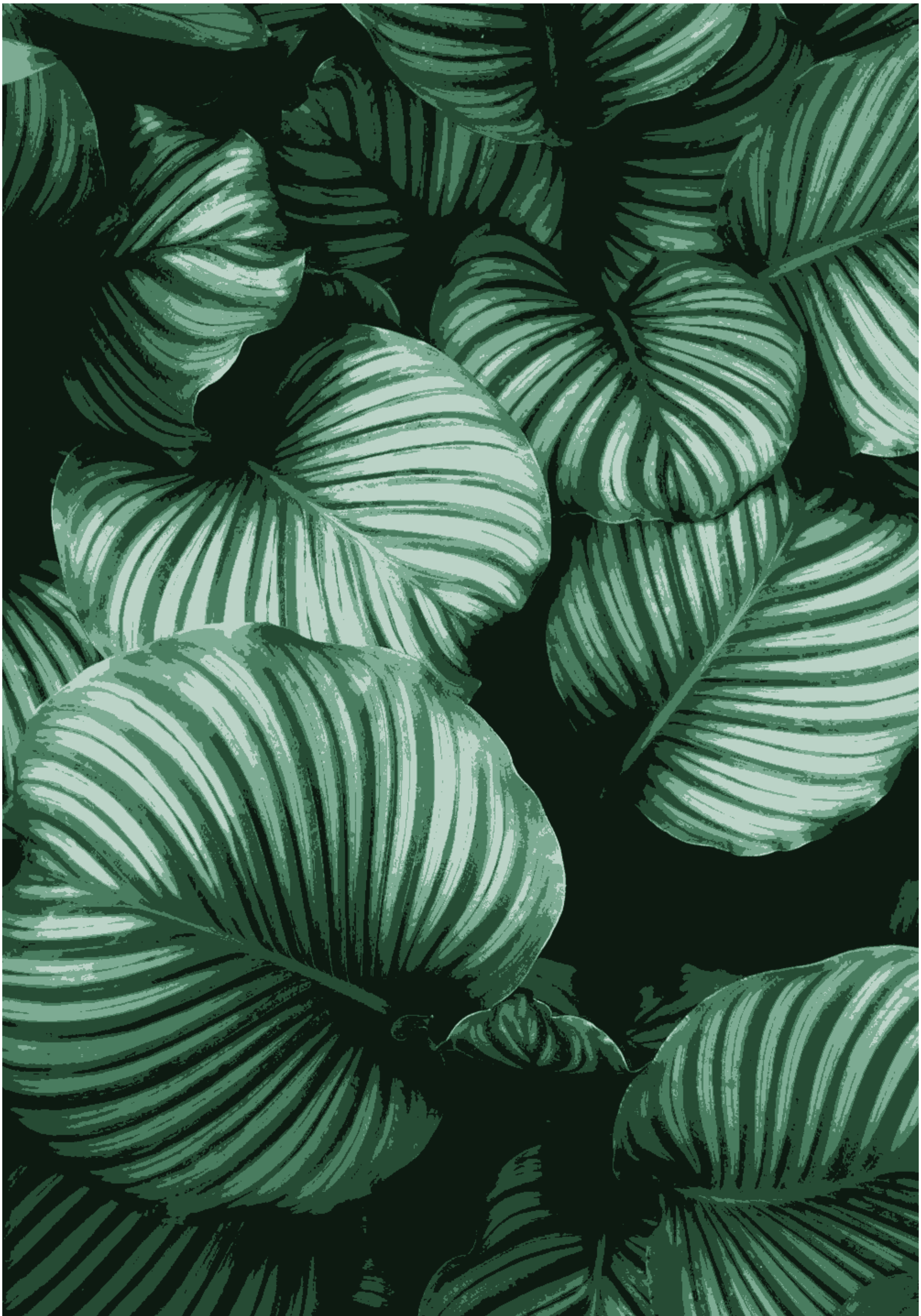


sloth – 20-means

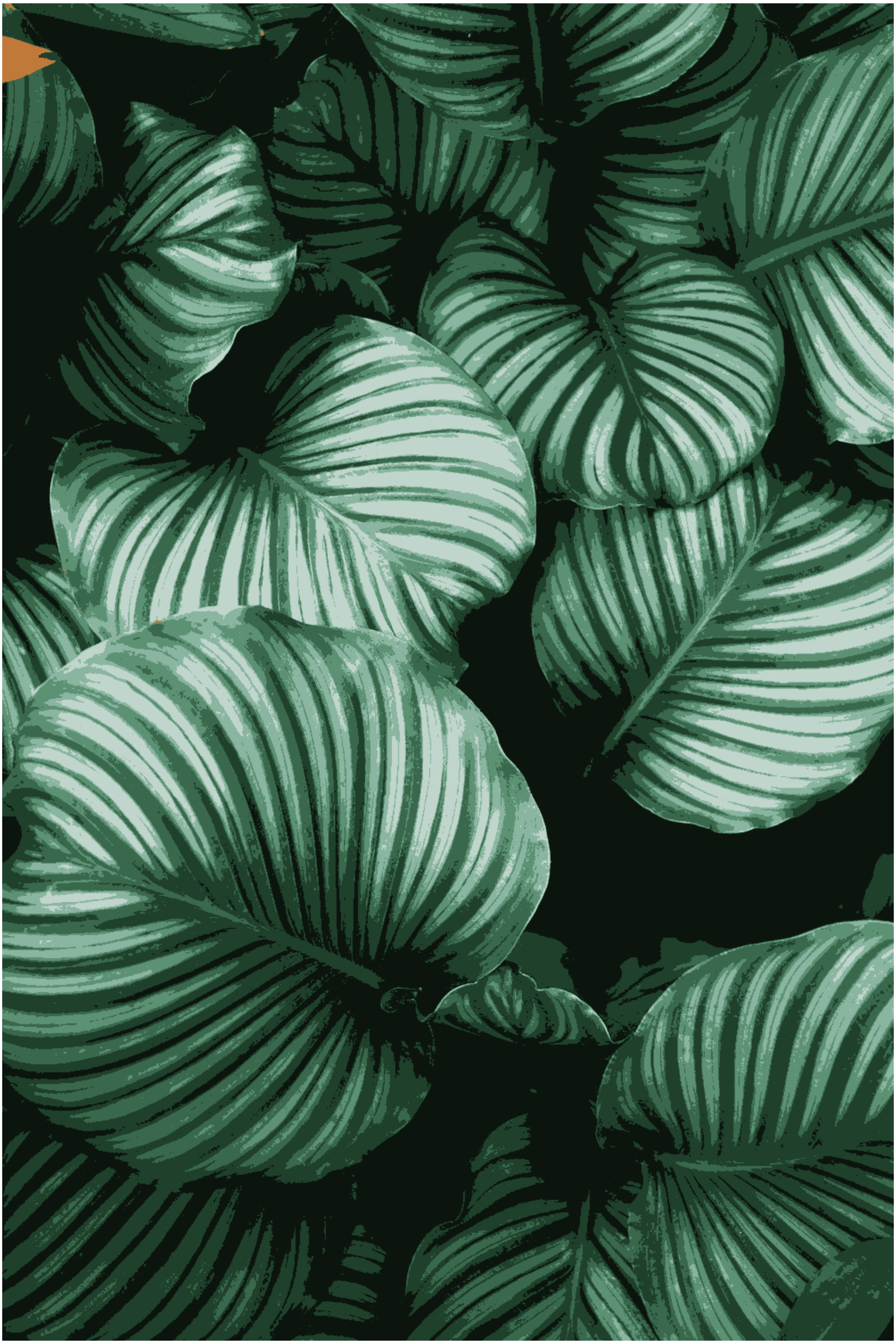


leaves – origin

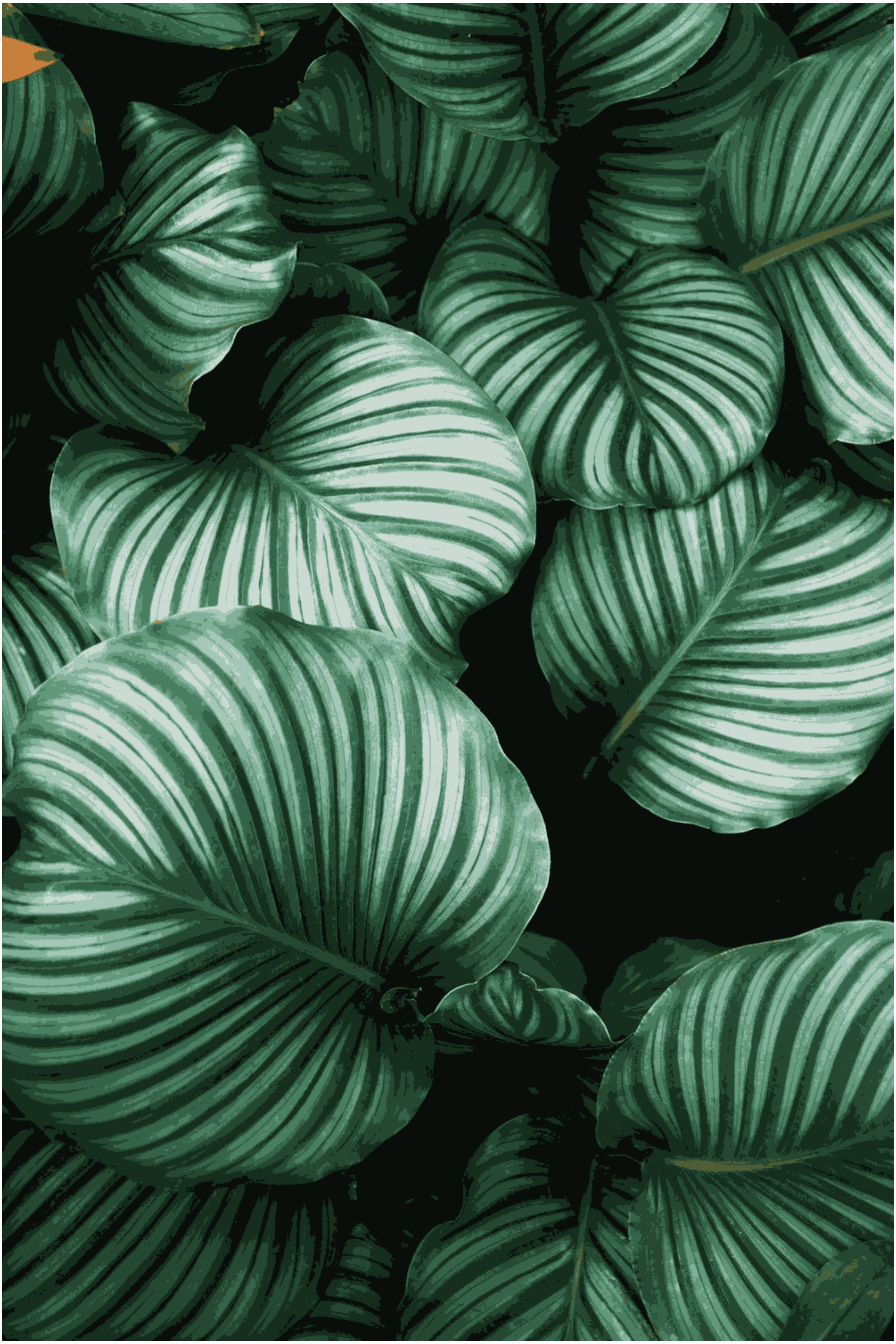
leaves – 5-means



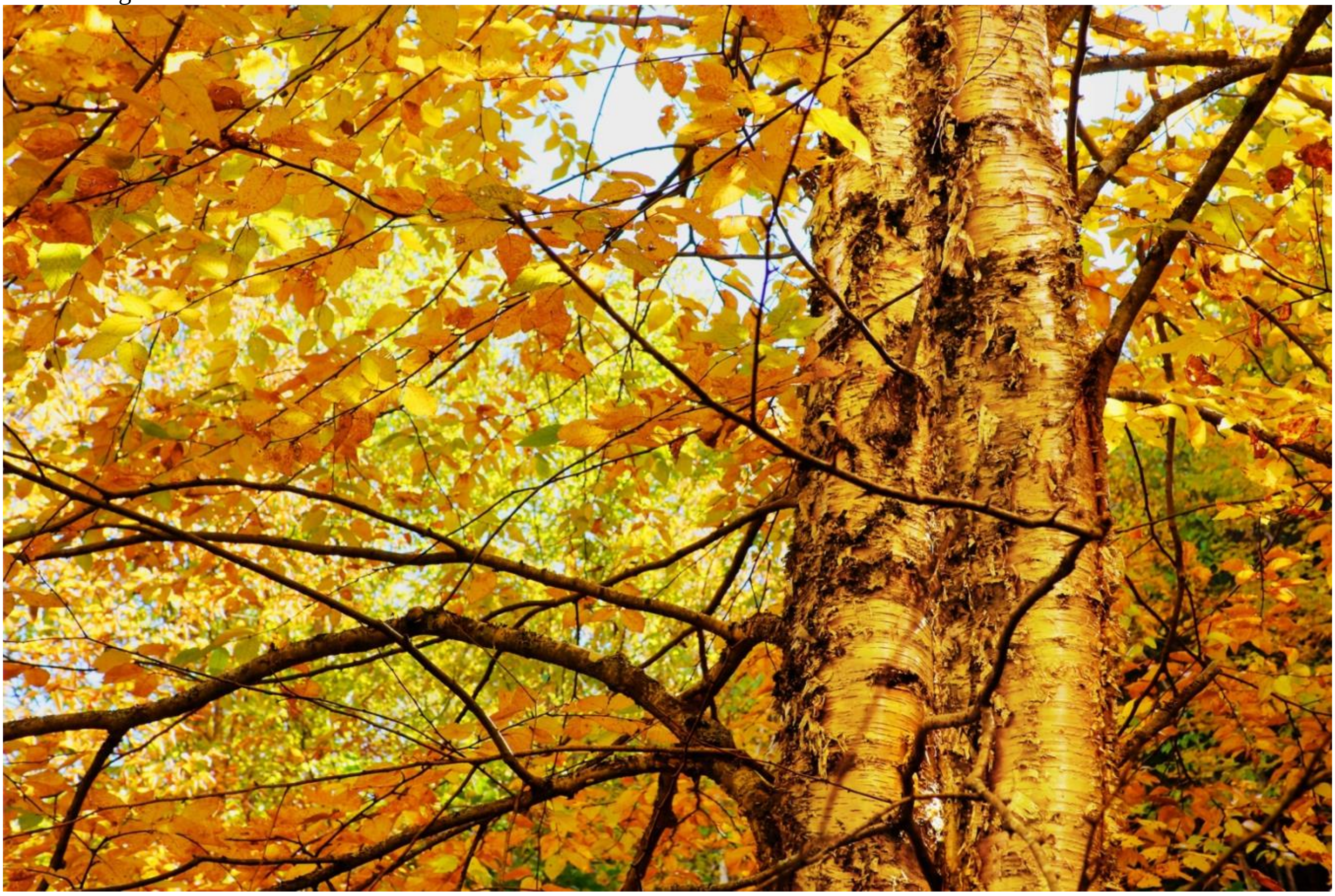
leaves – 10-means



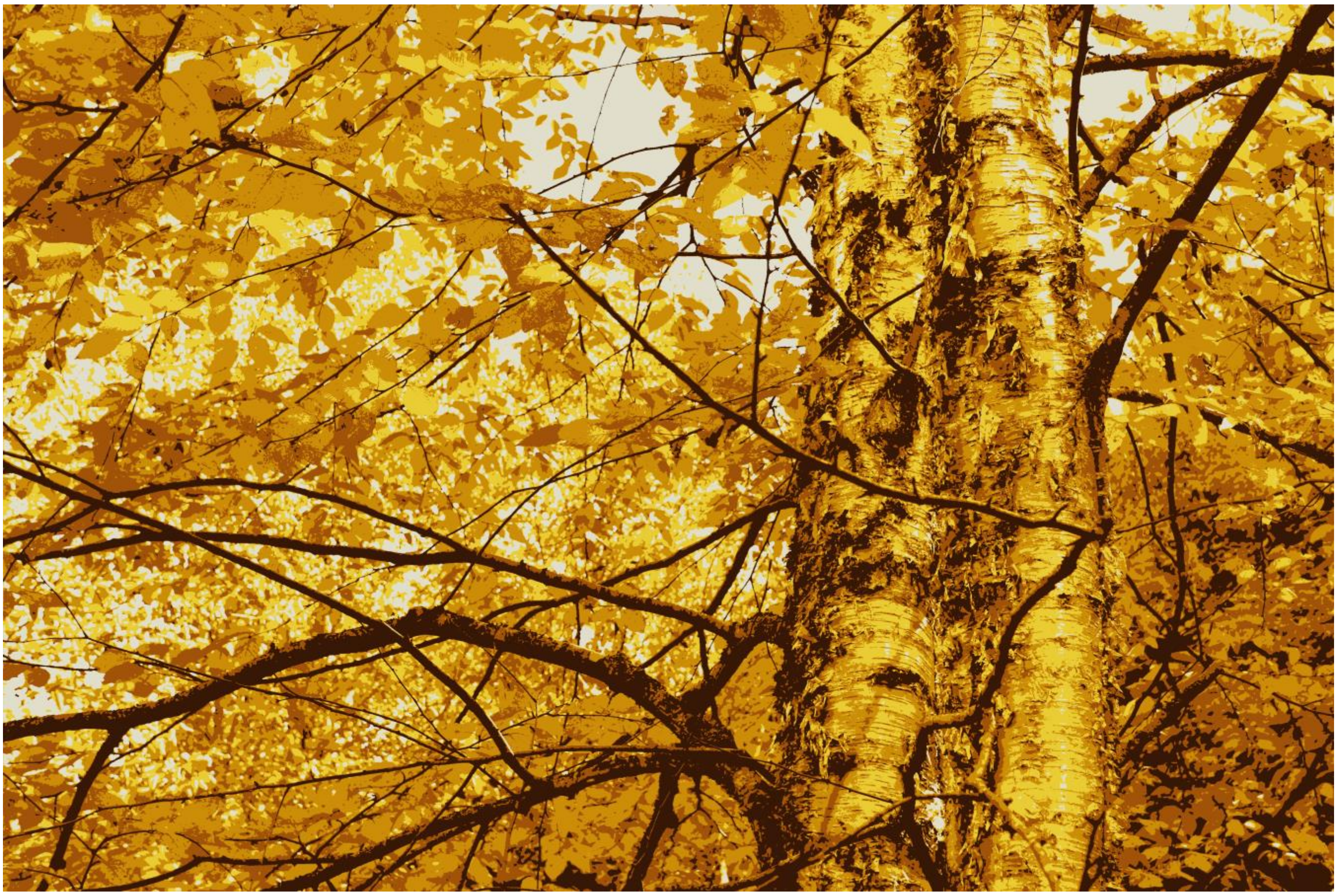
leaves – 20-means



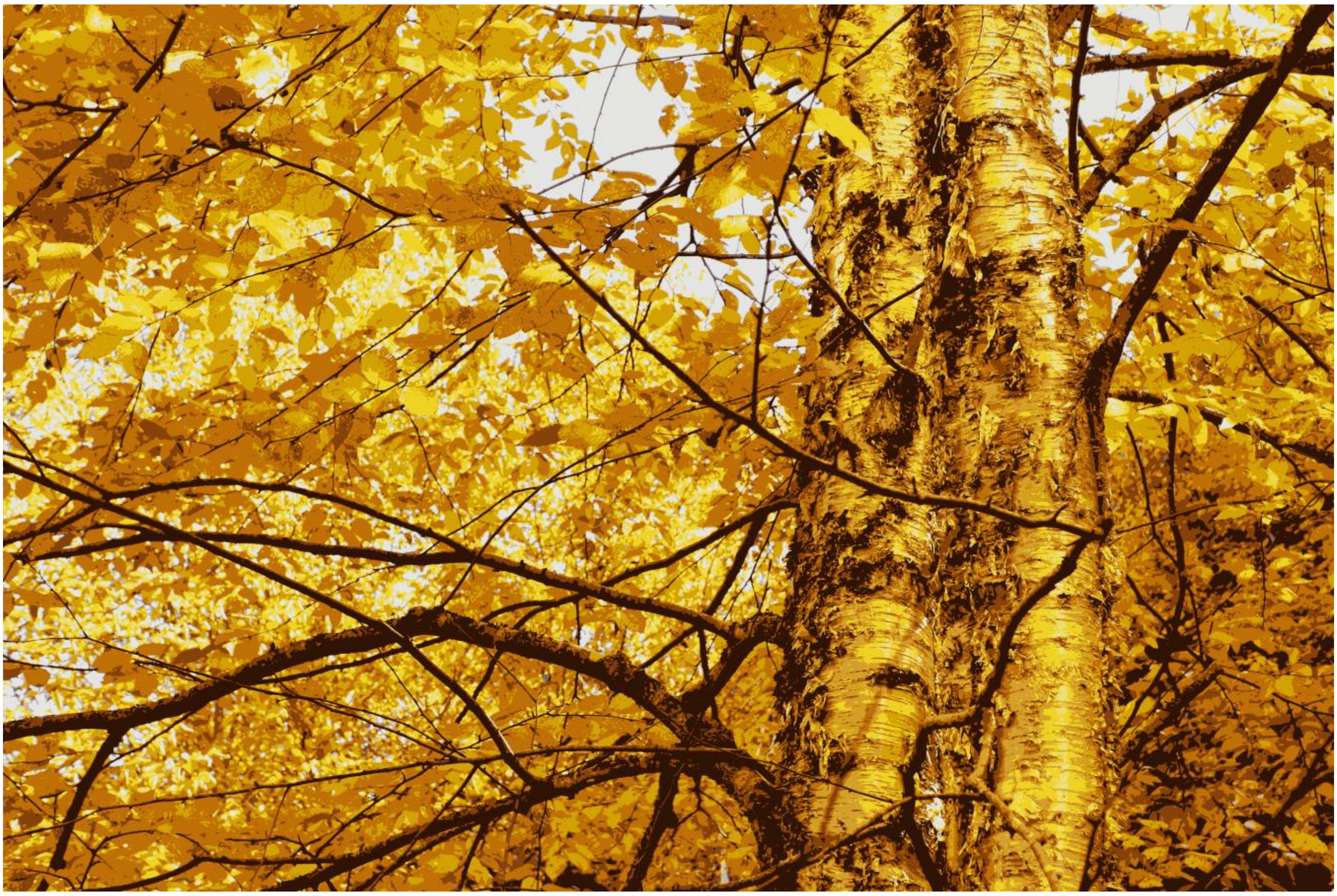
forest – origin



forest – 5-means



forest – 10-means



forest – 20-means