Bananalyzer

ADVANCED DATA STRUCTURES

BEATRIZ AGUIAR

ANDRÉ PEREIRA

JOÃO MARINHO

Timeline

01.

Research

How to use Octress for color quantization

02.

Eureka

Dmitry Alimov

https://github.com/delimitry/octree_color_quantizer

03.

Quantization improvements

04.

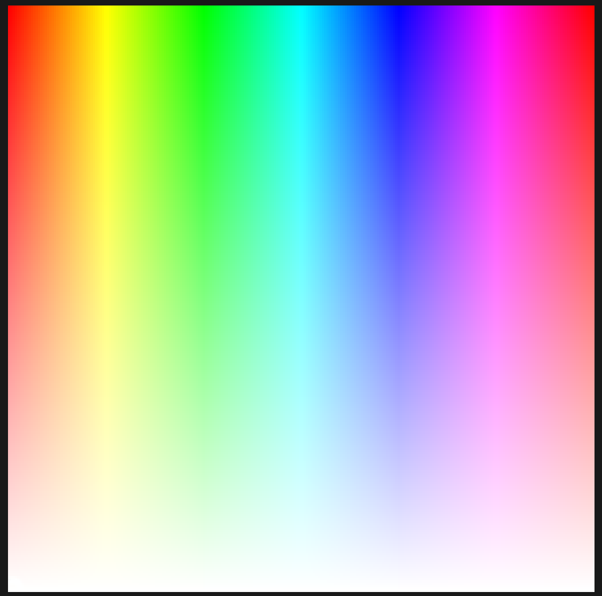
Jaccard Coefficient

05.

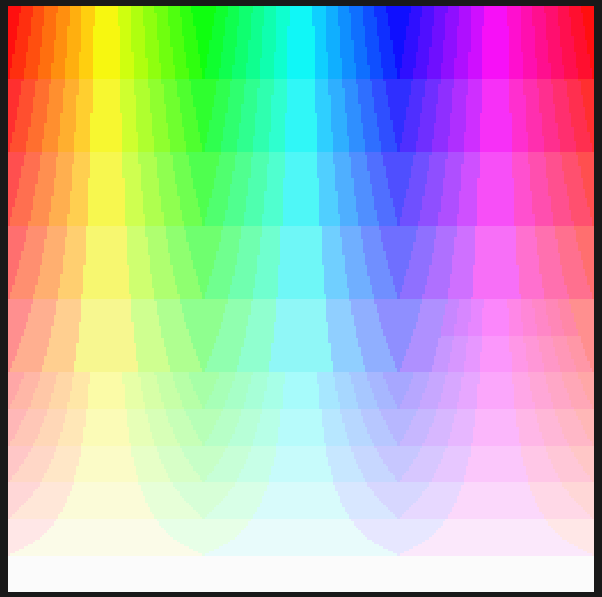
Image Processing improvements

06.

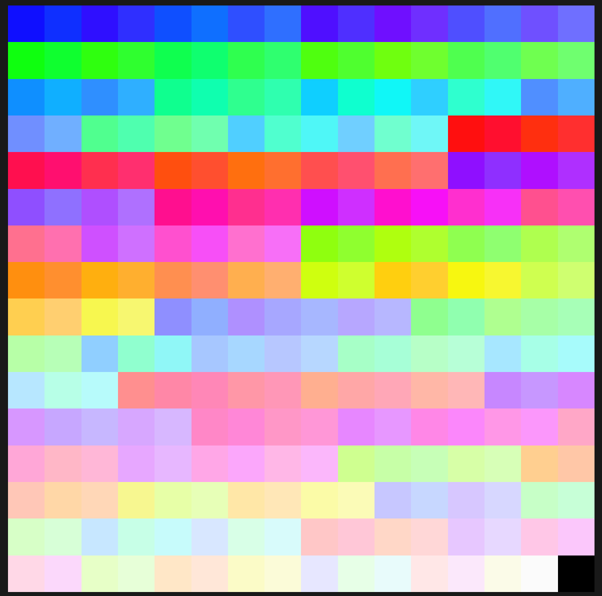
Results



[original image]



[image color palette]



[reduced image color palette]

Main Idea

Convert image to pixels

imageColors = Array(4096) [Color, Color, Color, Color, ...]

Create Octree from Color array

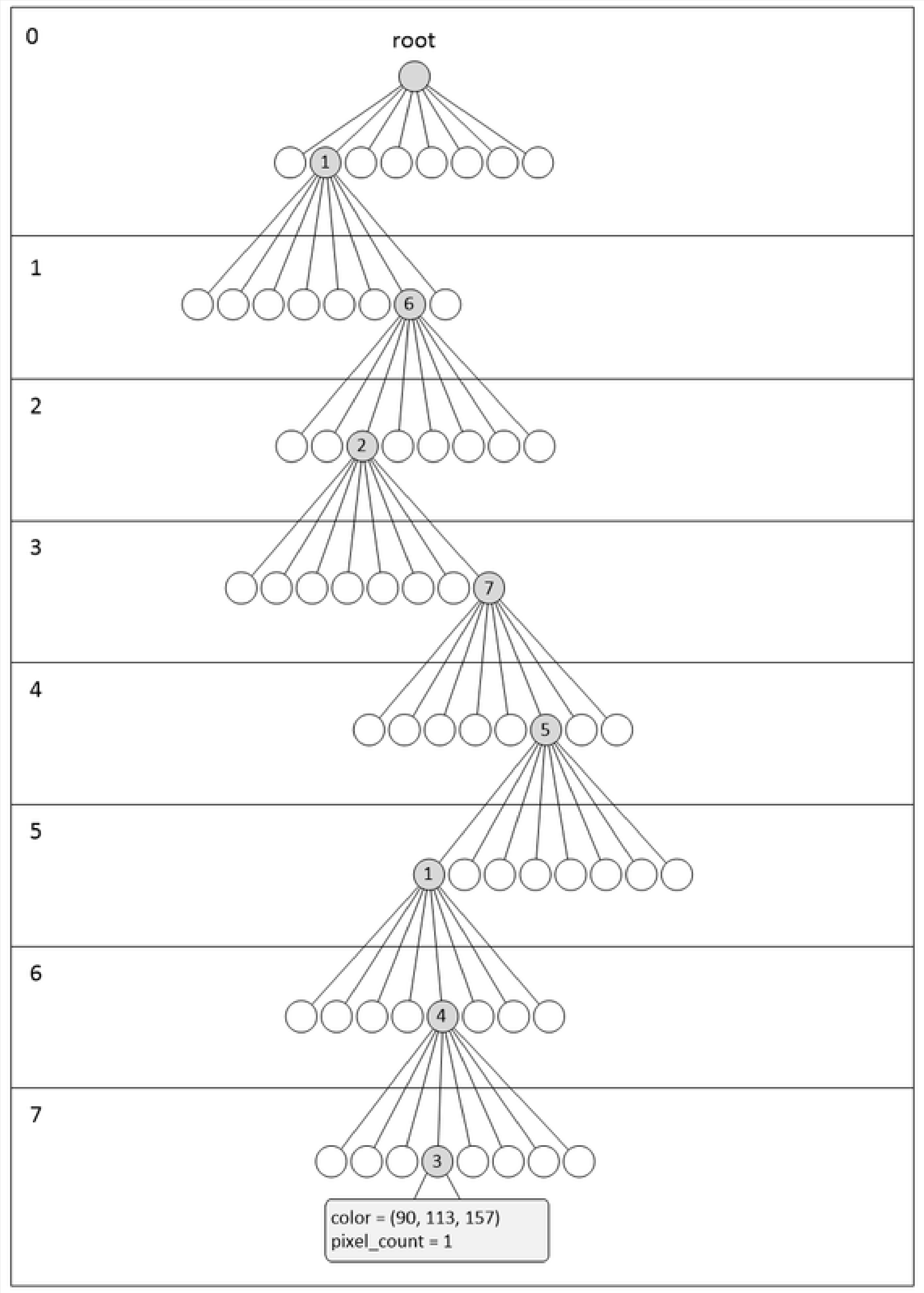
The idea of the tree is that each branch contains similar colors, so that we can 'bin' them together

Create Color Palette

Transverse through the tree from the bottom-up, 'folding up' and combining leaves until it fills the palette array with specified number of colors.

Color Octree

In the context of this explanation, the octree represents a hierarchical organization of colors.



01.

Structure

02.

Operations

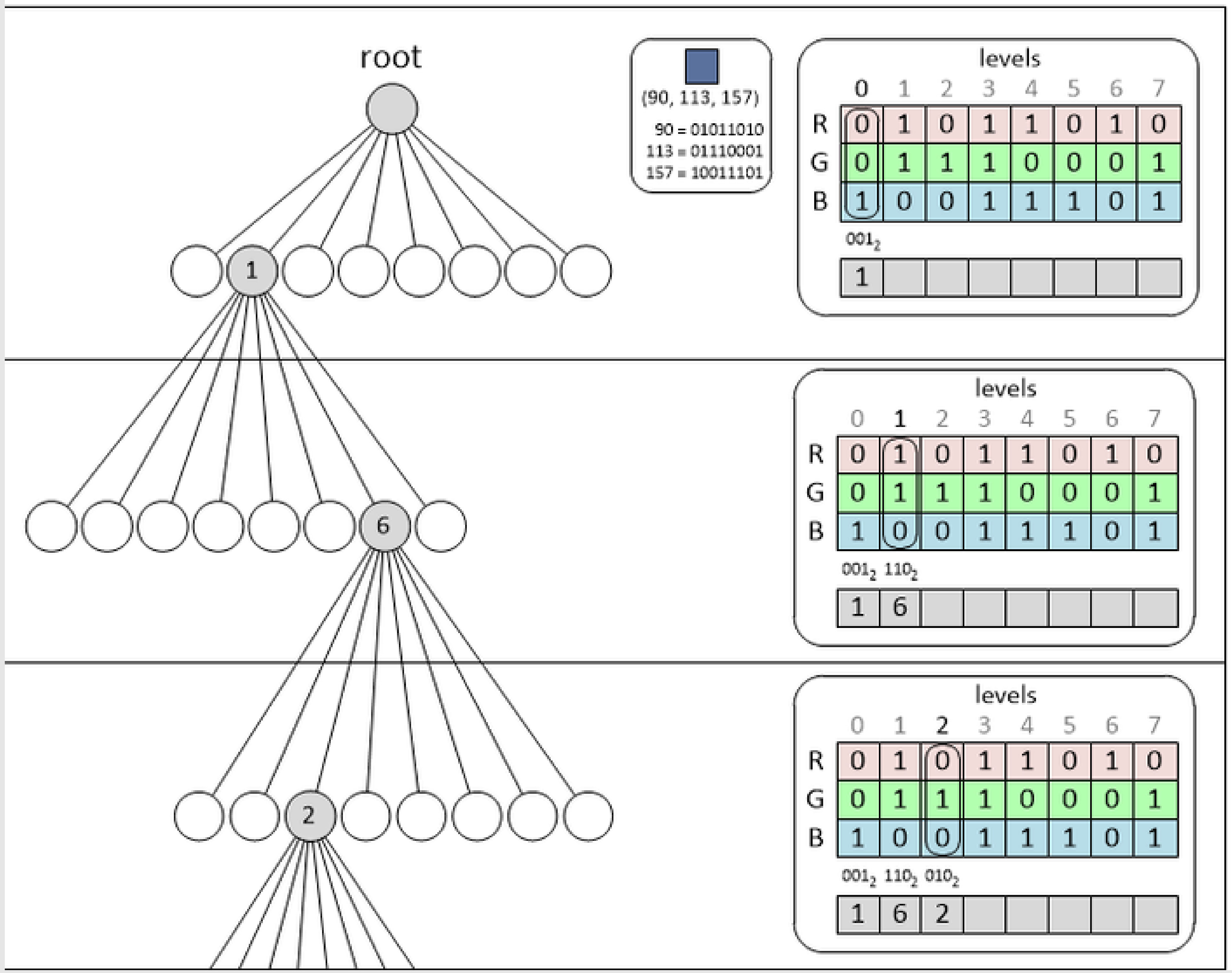
03.

Example

Structure

```
Node {
  id;
  color;
  pixel_count;
  level;
}
```

- In the 1st level, RGB is represented by 3 bits (0-7 colors)
- The following levels add one bit to each RGB component
- Colors are more complex the higher the level
- Maximum depth is 8



Operations

Insertion

O(n)

- An octree is a tree data structure where each internal node has exactly eight children.
- When inserting a new node (color) into the octree, the process begins by adding it to the root node of the tree. Then, it is added to all subsequent levels of the octree.
- The index of the child where the node will be placed in each level is calculated based on the binary representation of the color. The index ranges from 0 to 7.
- If there isn't a child node at the calculated index position, it creates a new node with the color of the current node and adds it as a child at that index position.
- If there is already a child node, the color of the current node is added to the existing node.

Identical colors are grouped together

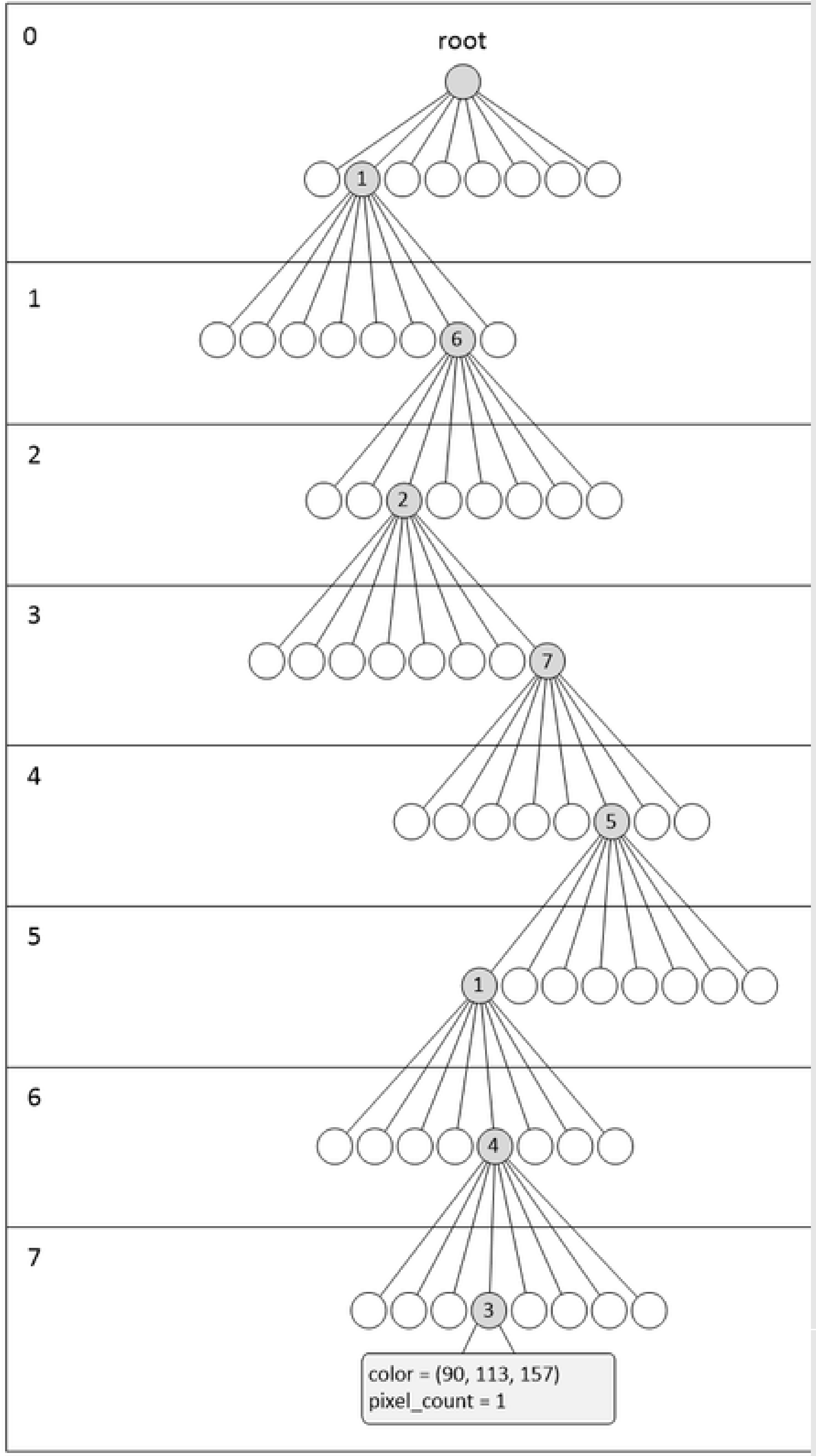
Make pallete

O(nlog(n))

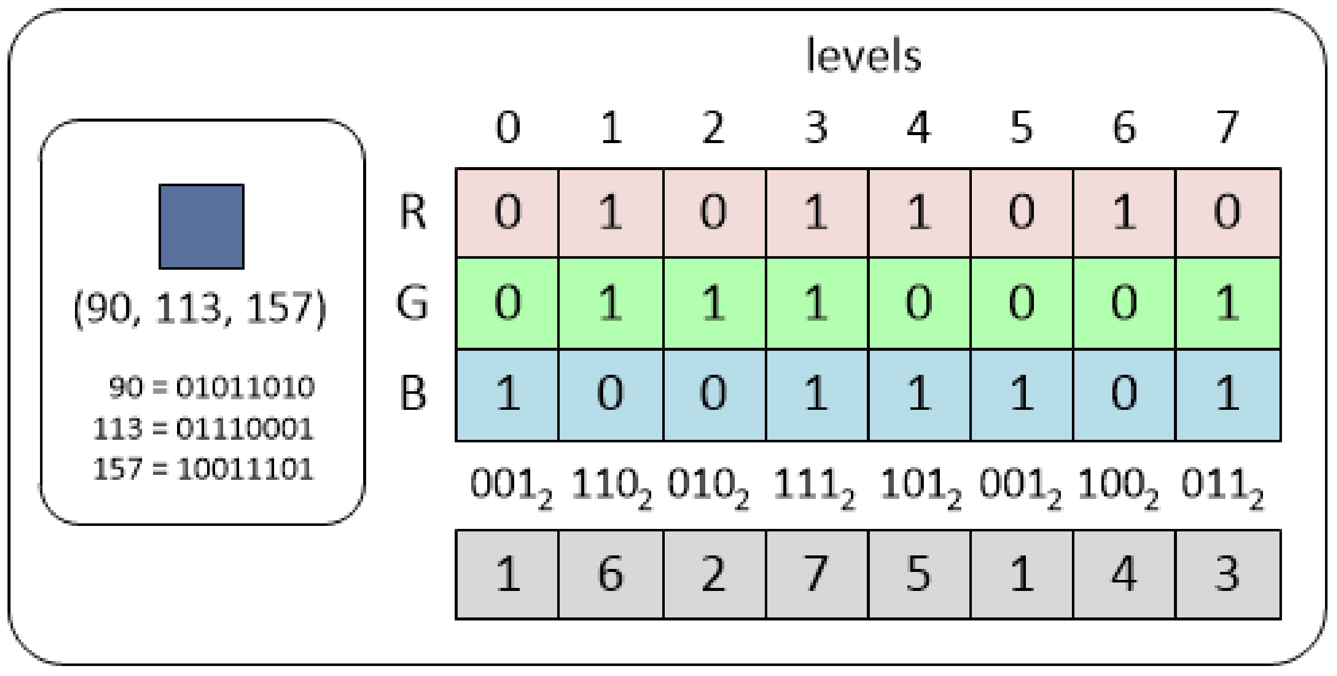
1. Transverse tree bottom up
2. Remove leaf nodes until *leaf_nodes = nr_of_palette_colors*
3. Nodes are removed in groups of 8 (children)
4. Add all children's pixels count and color channels to the parent node
5. Nodes with less pixel count are removed first

The palette is filled with average colours, from each leaf. As each leaf has the number of pixels with color and color's sum of R, G and B values,
palette_colour = (color.R / pixel_count, color.G/ pixel_count, color.B / pixel_count).

Example



color = (90, 113, 157)
pixel_count = 1

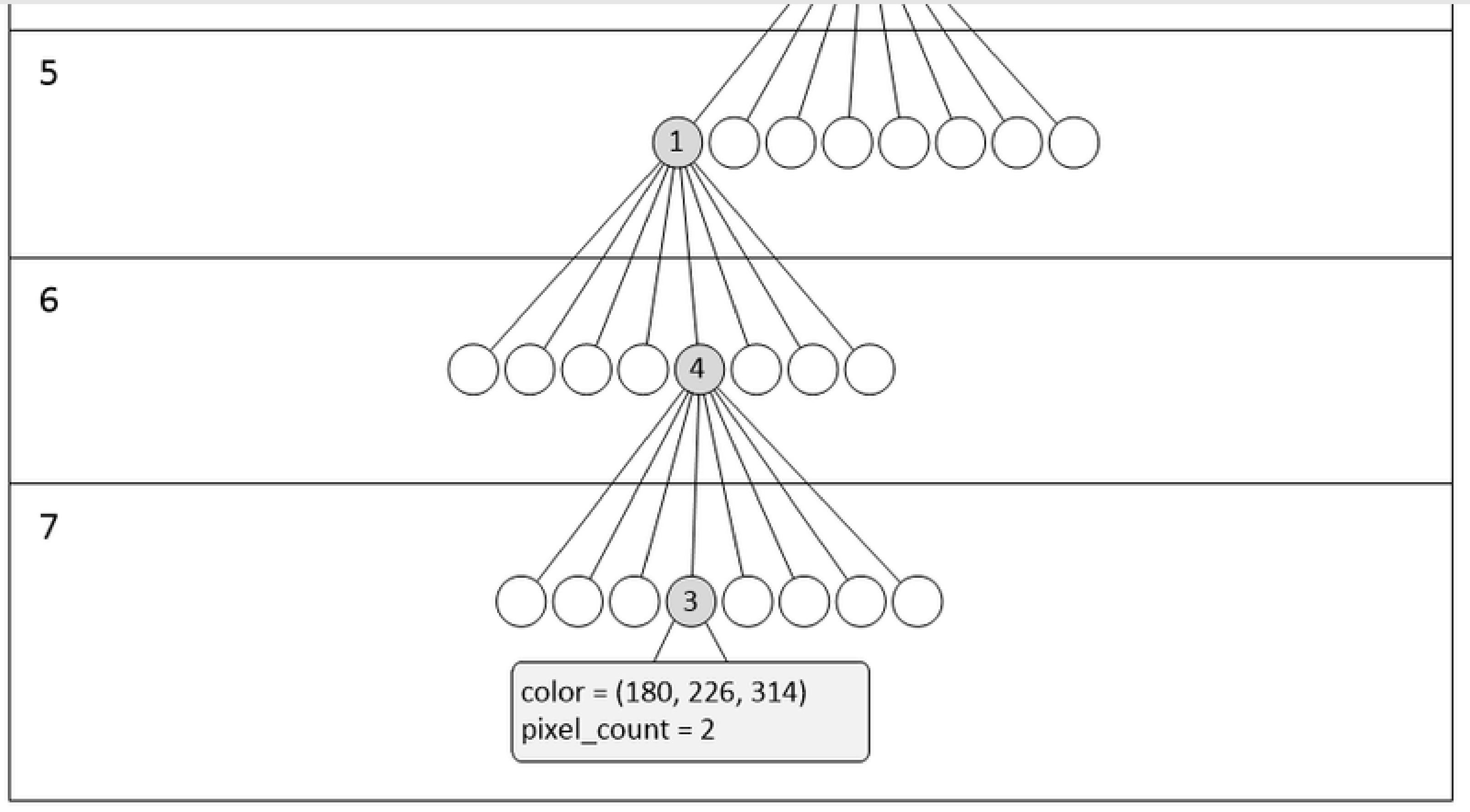


(90, 113, 157)

90 = 01011010
113 = 01110001
157 = 10011101

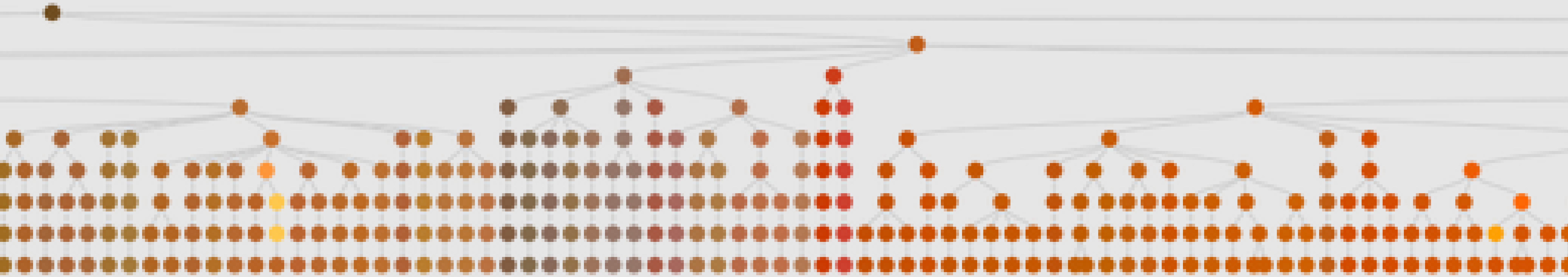
levels

	0	1	2	3	4	5	6	7
R	0	1	0	1	1	0	1	0
G	0	1	1	1	0	0	0	1
B	1	0	0	1	1	1	0	1
	001_2	110_2	010_2	111_2	101_2	001_2	100_2	011_2
	1	6	2	7	5	1	4	3



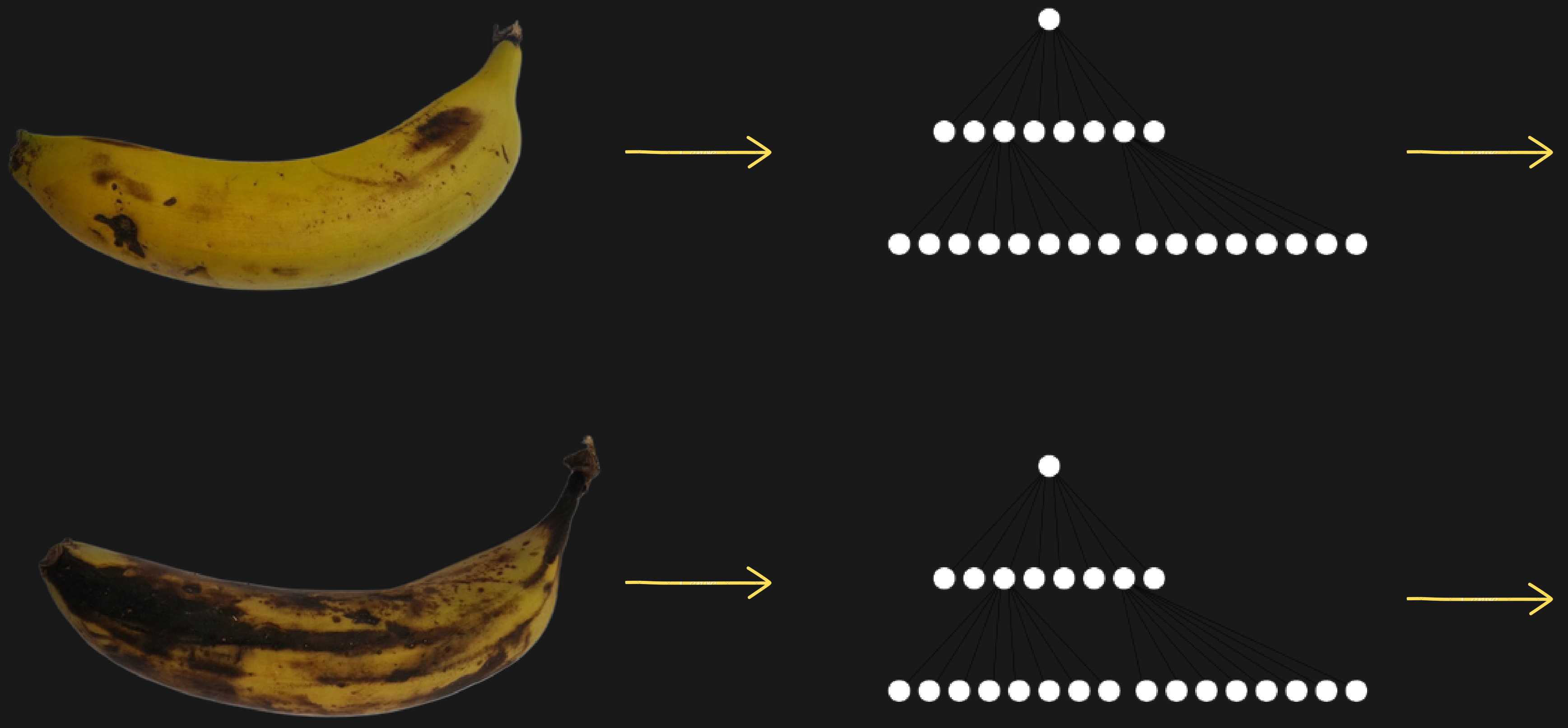
color = (180, 226, 314)
pixel_count = 2

Example



Jaccard

O(n)



{ ... }

set of node ids

{ ... }

set of node ids

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

```
intersection = set1.intersection(set2)
union_len = len(set1) + len(set2) - len(intersection)
jaccard = len(intersection) /union_len
```

Testing

Dataset

104 unripe banana images
48 barely ripe banana images
88 ripe banana images
33 overripe banana images

source: Saranya, N., Srinivasan, K. & Kumar, S.K.P. Banana ripeness stage identification: a deep learning approach. J Ambient Intell Human Comput 13, 4033–4039 (2022).

- same lighting
- background removed
- resized image

Testing

- Train test split
  - 20% testing data
  - 80% sampling data
- Depth variation
  - [4, 6, 8]
- Palette size variation
  - [64, 128, 256, 512]

Results reliability

- accuracy
- prediction time

Processing Image improvements

01.

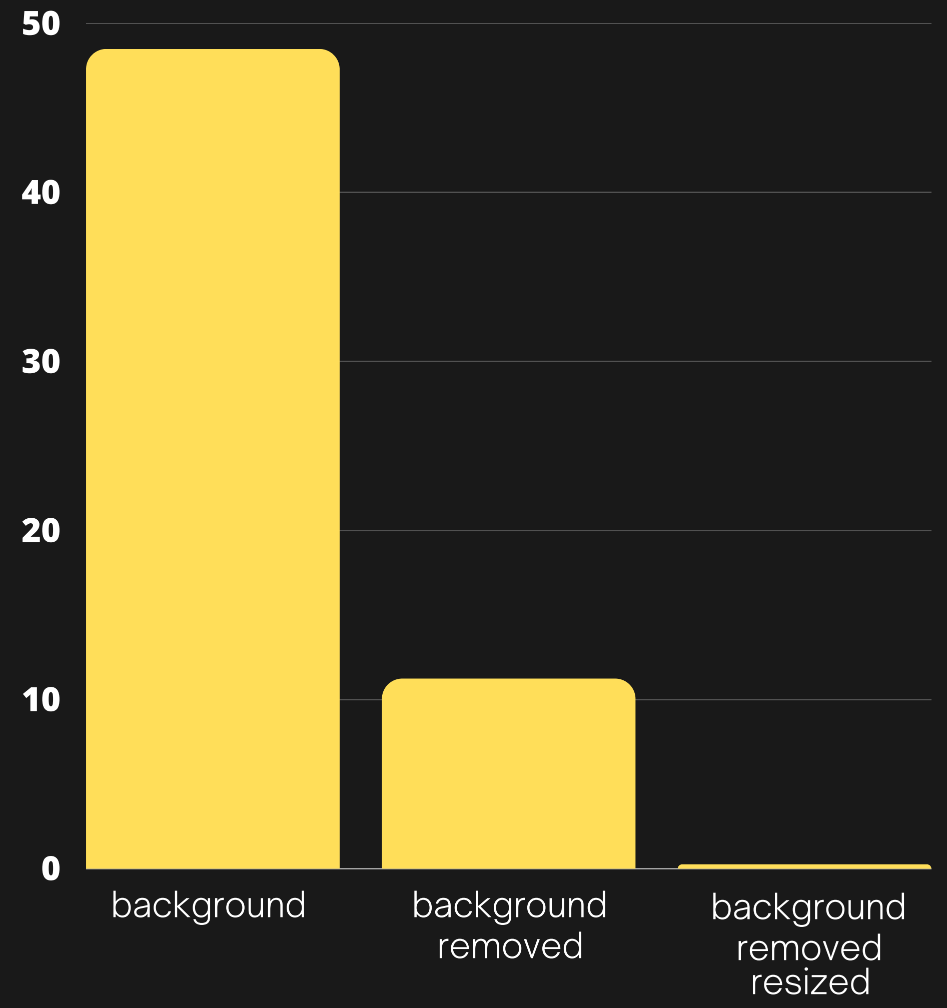
Remove background

- Used python library rembg.
- Pixels with alpha < 1 are not added to the octree.

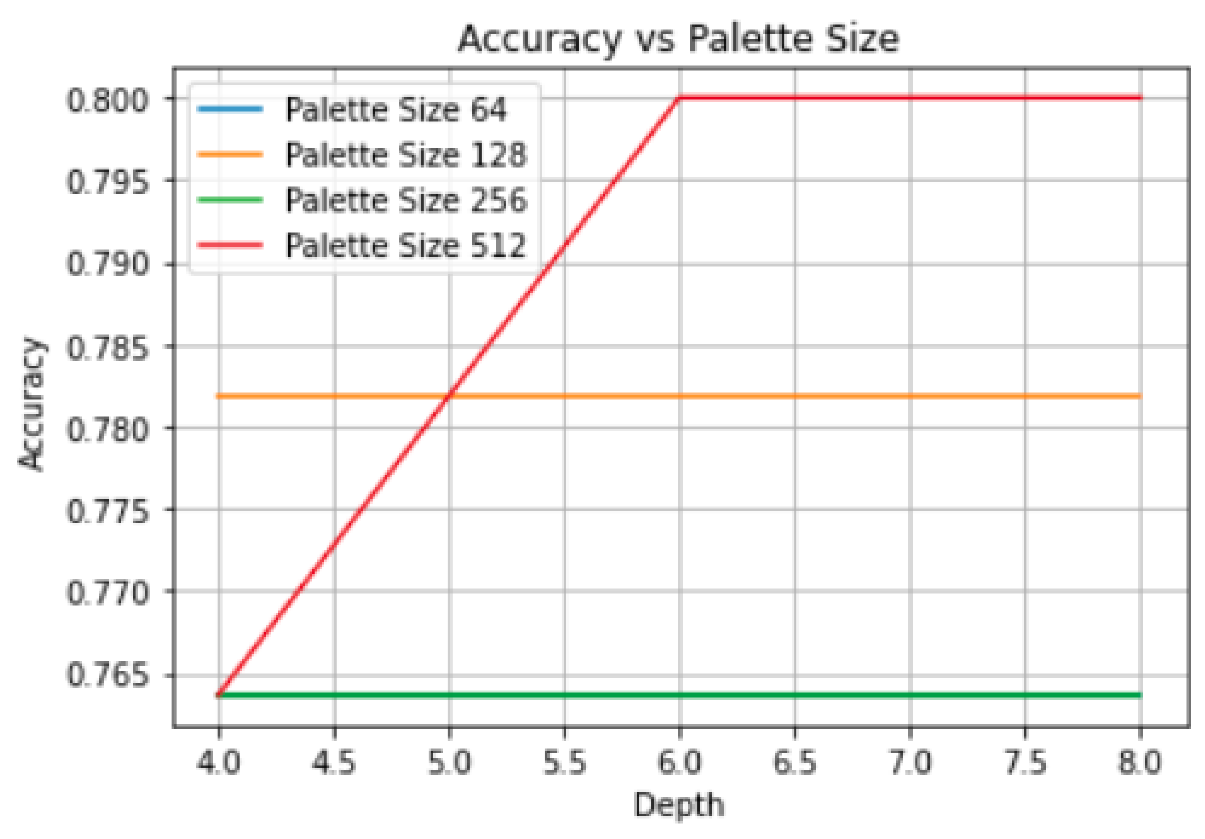
02.

Image resizing

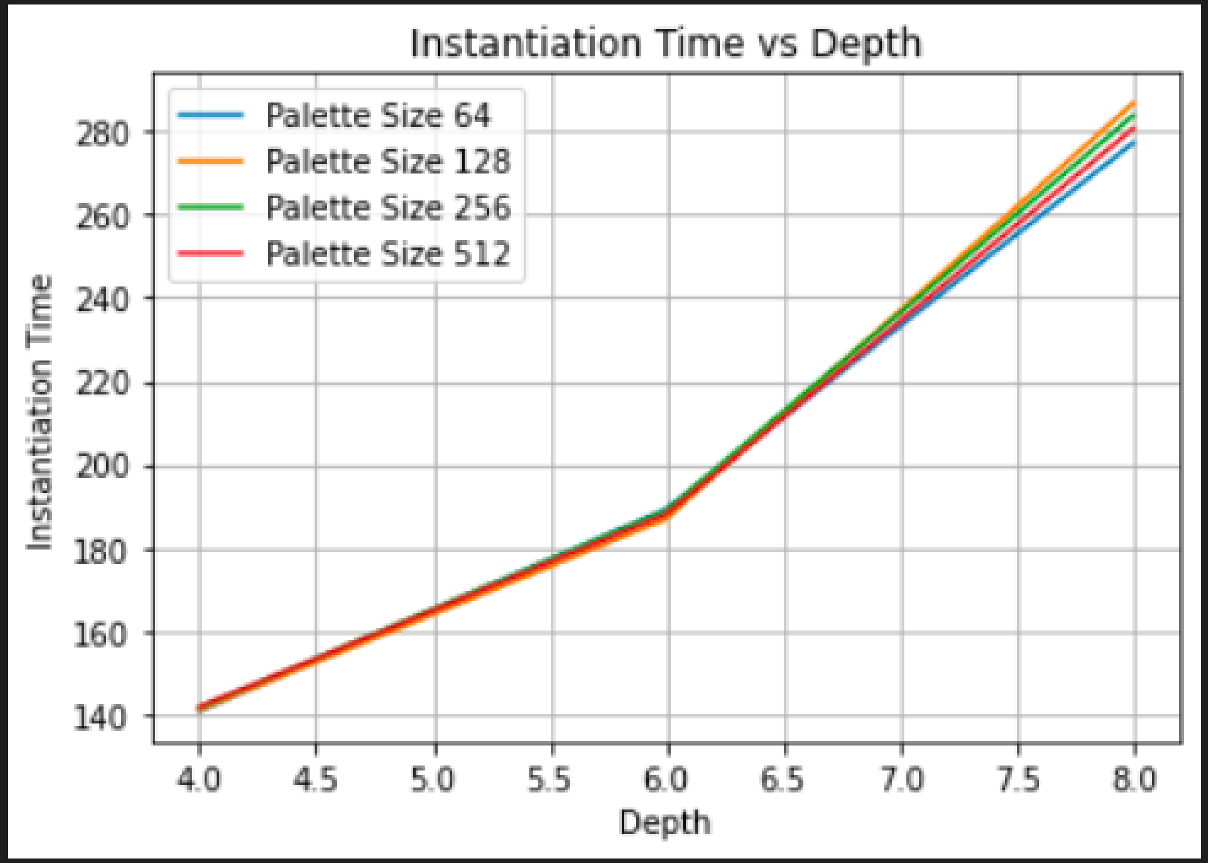
https://neurohive.io/en/popularnetworks/vgg16/



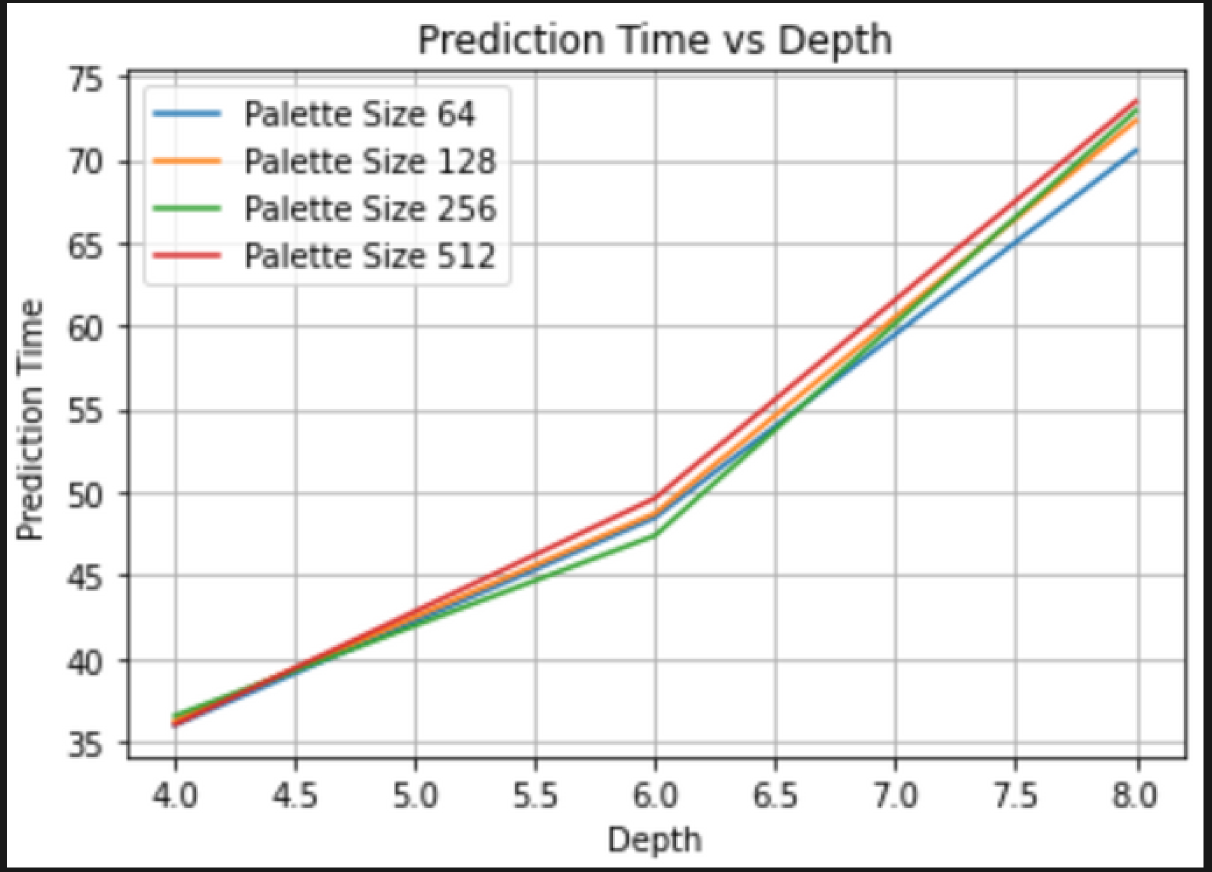
Results



Accuracy vs Palette Size
Palette Size 64
Palette Size 128
Palette Size 256
Palette Size 512
Accuracy
0.800
0.795
0.790
0.785
0.780
0.775
0.770
0.765
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Depth



Instantiation Time vs Depth
Palette Size 64
Palette Size 128
Palette Size 256
Palette Size 512
Instantiation Time
280
260
240
220
200
180
160
140
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Depth



Prediction Time vs Depth
Palette Size 64
Palette Size 128
Palette Size 256
Palette Size 512
Prediction Time
75
70
65
60
55
50
45
40
35
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Depth

Is it reliable?

Parameters:

- 512 palette size
- 6 depth

Load sample dataset:

- 273 labeled images
- background removed
- resized
- 240.1258 seconds instantiation time

Load single image:

- background removed
- resized
- 0.8703 seconds instantiation time

Predict ripeness level:

- 0.0052 seconds prediction

Total prediction time	**0.8755 seconds**
Accuracy	**0.8**

30 banana images (all types of ripeness)

Questions?

BEATRIZ AGUIAR

ANDRÉ PEREIRA

JOÃO MARINHO