Skoltech
Skolkovo Institute of Science and Technology

Digital Agriculture Laboratory

Computer Vision in Digital Agriculture:
CV and Deep Learning techniques for apple segmentation

PI: Mariia Pukalchik
TA: Nikita Stasenko

Introduction to Digital Agriculture

Seminar

Term 3, 2020 - 2021

Skoltech
Skolkovo Institute of Science and Technology

Seminar Plan

1. Semantic segmentation
2. RGB and HSV colour models
3. Practice
4. Project Proposal

Semantic segmentation

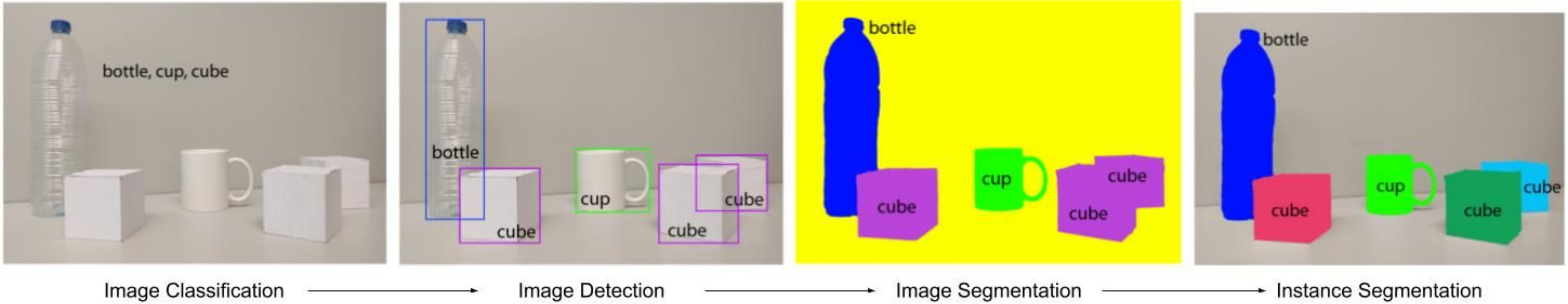
General Terms

Computer Vision (CV) – Methods and techniques through which artificial vision systems can be developed in a reasonable way in practical applications.

Image segmentation – The process of partitioning an image into multiple segments (sets of pixels, or just image objects).

Semantic segmentation – The task of clustering parts of an image together which belong to the same object class.

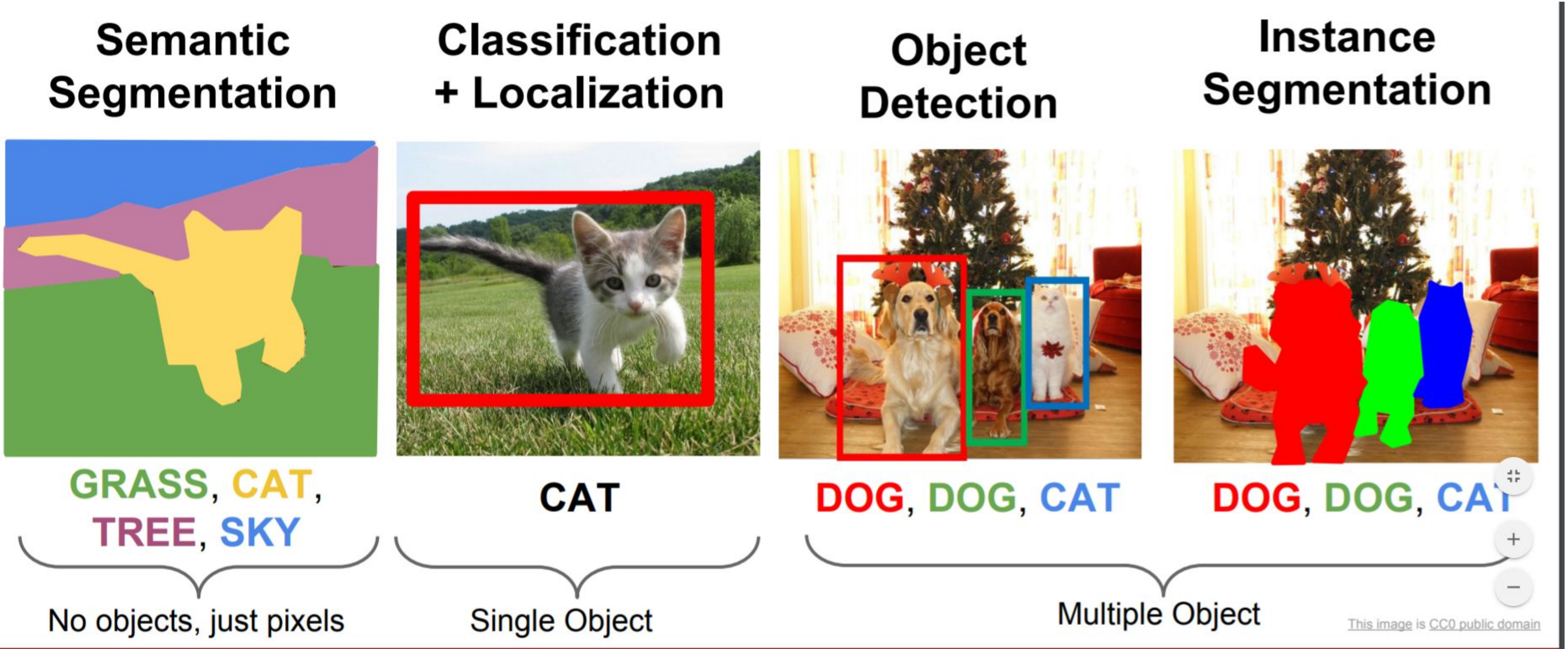
A brief history of semantic segmentation



Comprehending an entire image as a whole. **The goal is to classify the image by assigning it to a specific label.**	**Identifying of objects in the image.** **This step is close to the human level of image processing.**	**“Understanding” of image in pixel level.** **More close to the human level of image processing.**	**Detecting and delineating each distinct object of interest in an image.** **The prediction of object instances and their per-pixel segmentation mask.**

by https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic-segmentation-95082b025571

A brief history of semantic segmentation

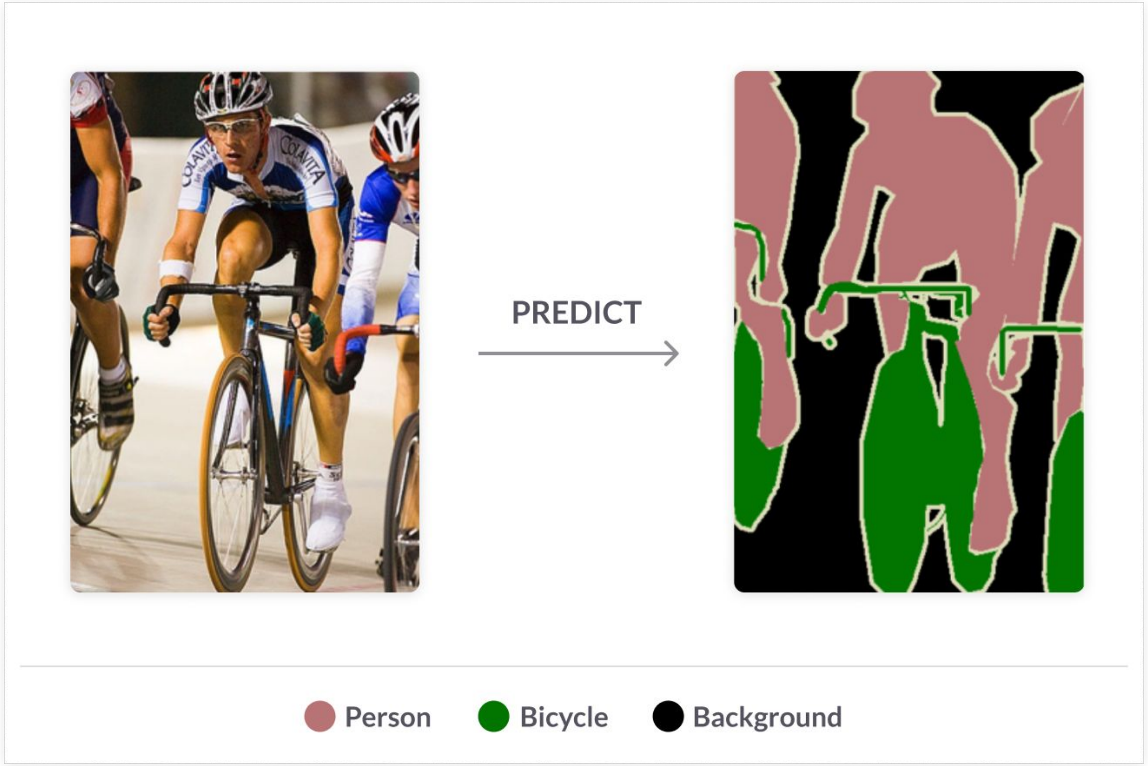


by https://data-flair.training/blogs/image-segmentation-machine-learning/

Semantic segmentation

Once again, semantic segmentation is a technique of segmenting image with “understanding” of image in pixel level.

It is a form of pixel-level prediction because each pixel in an image is classified according to a category.



by https://missinglink.ai/guides/computer-vision/image-segmentation-deep-learning-methods-applications/

Skoltech
Skolkovo Institute of Science and Technology

How computers understand colours

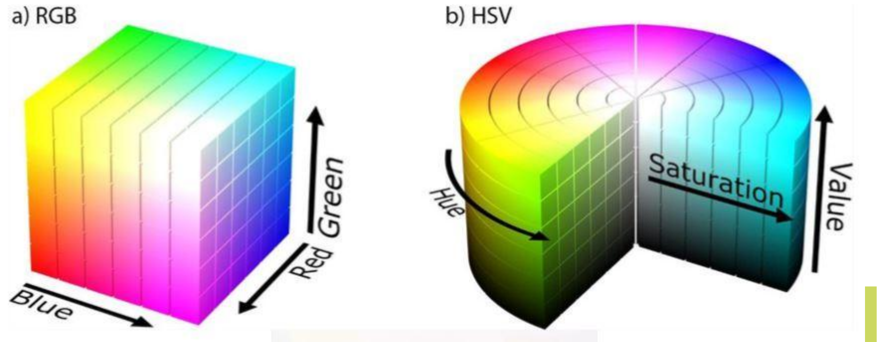
- The smallest number of image is a **pixel**, or a picture/colour element. An image contains multiple pixels arranged in rows and columns.

- Computers don't know colours, but only just numbers. So computer needs a **colour model** to convert colour elements to numbers.

- **Colour model** is an **abstract mathematical model** that describes how **colours** can be represented as a **set of numbers**.

The most applied colour models:

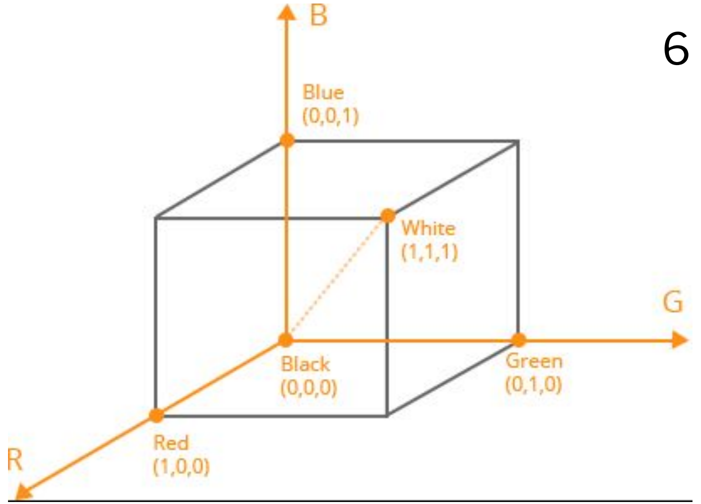
1. **RGB (Red, Green, Blue) colour model;**
2. **HSV (Hue, Saturation, Value) colour model;**

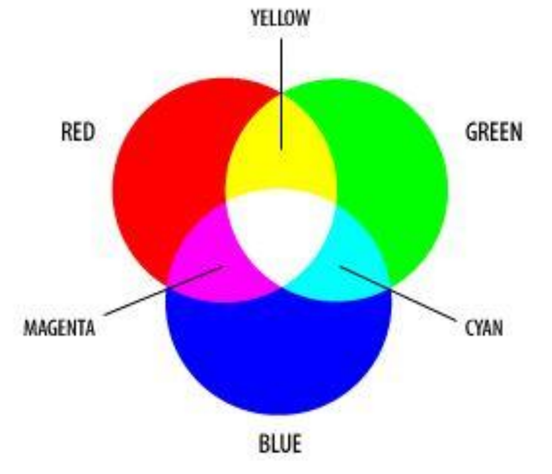


RGB and HSV colour models

RGB colour model

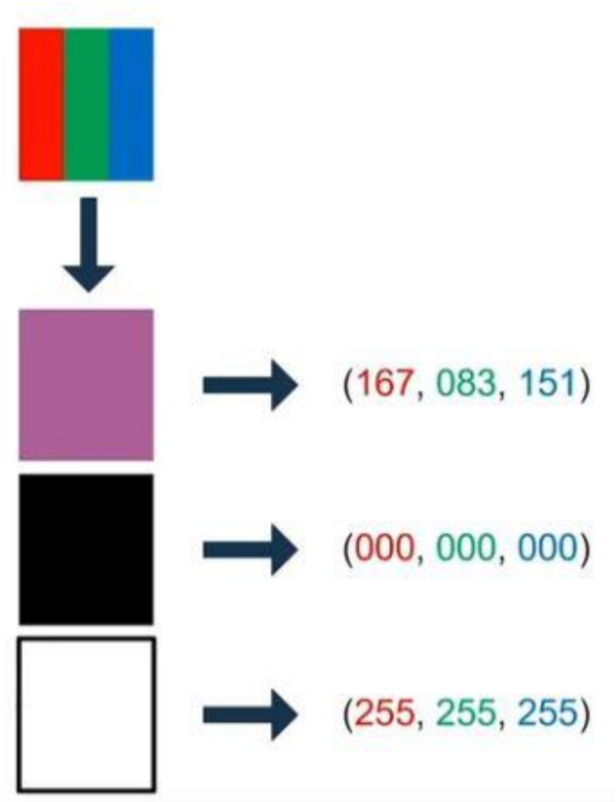
- RGB is an additive colour model.
- Each colour is composed by the 3 primary additive colours - **Red**, **Green**, **Blue**. It is based on a cartesian coordinate system [0...1].
- Each pixel is represented by three numbers .





RGB colour model

- For each primary colour, it is possible to take 256 different shades of that colour.
- By adding 256 shades of 3 primary colours it is possible to produce over 16 million different colours.
- In grayscale (balck and white) images, each pixel is a single number, representing the amount of light in range from 0 (black) to 255 (white).



RGB colour model

Advantages:

1. No transformations required to show data on the screen.
2. It is considered as the base colour space for various applications.
3. It is computationally practical system.
4. This model is very easy to implement.

Limitations:

1. RGB values are commonly not transferable between devices.
2. Not perfect for colours identification.
3. Difficult to determine specific colour.
4. Difference between colours is not linear.

HSV colour model

- **HSV** is a cylindrical coordinate representation of points in an RGB colour model.

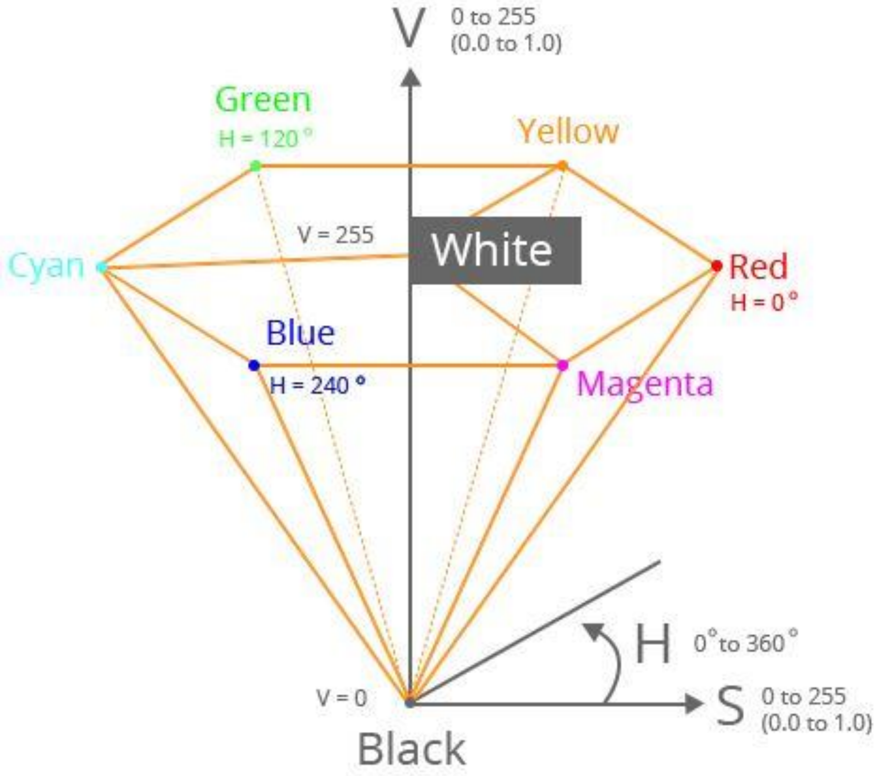
- **Hue (H)** - the dominant colour as perceived by an observer. Top surface, showing the change in hue in the H direction from 0 to 360 (it is the entire spectrum of visible light).

- **Saturation (S)** - the amount of white light mixed with a Hue. Represented by S direction from the center to the hexagonal boundary, varying from 0 to 1 values.

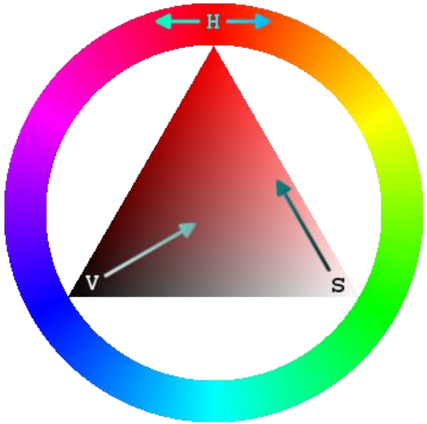
- **Value (V)** - the chromatic notion of intensity. V denotes the height of the hexagonal pyramid and represents a black to white gradation from bottom to top (0 - 1).

HSV colour model

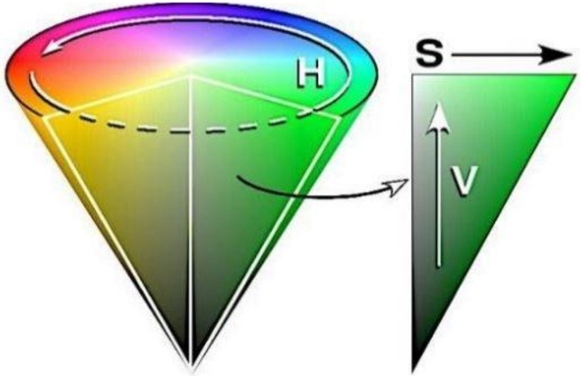
- **(H)** - the dominant colour as perceived by an observer. Top surface, showing the change in hue in the H direction from 0 to 360 (it is the entire spectrum of visible light).
- **(S)** - the amount of white light mixed with a Hue. Represented by S direction from the center to the hexagonal boundary, varying from 0 to 1 values.
- **(V)** - the chromatic notion of intensity. V denotes the height of the hexagonal pyramid and represents a black to white gradation from bottom to top (0 - 1).



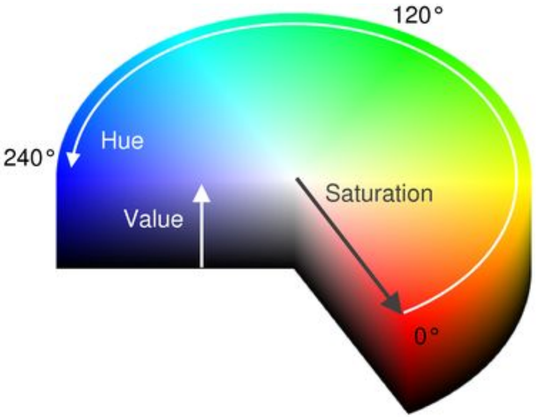
HSV colour model visualizations



Ring form



Cone form



Cylinder form

HSV colour model

Advantages:

1. **Can be clearly defined by human perception.**
2. **The chromaticity is decoupled from intensity.**
3. **Robust before non-uniform illumination.**

Limitations:

1. **Non removable signatures.**
2. **Analysis requires the use of circular statistics.**
3. **Because H is a circular quantity, represented numerically with a discountinity of 360, it is difficult to use statistical computations or quantitative comparisons**

Converting RGB to HSV

$$R = \frac{R'}{\text{scale}_r}, G = \frac{G'}{\text{scale}_g}, B = \frac{B'}{\text{scale}_b} \quad (1)$$

$$m_{\max} = \max(R, G, B) \quad (2)$$

$$m_{\min} = \min(R, G, B) \quad (3)$$

$$\Delta = m_{\max} - m_{\min} \quad (4)$$

$$(5)$$

$$H = \begin{cases} \text{undefined}, & \text{if } \Delta = 0 \\ \frac{G-B}{\Delta} & \text{if } m_{\max} = R \\ \frac{B-R}{\Delta} + 2 & \text{if } m_{\max} = G \\ \frac{R-G}{\Delta} + 4 & \text{if } m_{\max} = B \end{cases} \quad (6)$$

$$H' = H \times \text{scale}_h \quad (7)$$

by https://mattlockyer.github.io/iat455/documents/rgb-hsv.pdf

Converting RGB to HSV

$$V = m_{\max} \quad (8)$$

$$V' = V \times \text{scale}_v \quad (9)$$

$$(10)$$

$$S = \begin{cases} 0, & \text{if } V = 0 \\ \frac{\Delta}{V} & \text{otherwise} \end{cases} \quad (11)$$

$$S' = S \times \text{scale}_s \quad (12)$$

by https://mattlockyer.github.io/iat455/documents/rgb-hsv.pdf

Converting HSV to RGB

$$H = \begin{cases} \text{undefined} & \text{if } H' \text{ is undefined} \\ (\frac{H'}{\text{scale}_h} \bmod 6) + 6 & \text{if } H' < 0 \\ \frac{H'}{\text{scale}_h} \bmod 6 & \text{otherwise} \end{cases} \quad (13)$$

$$S = \frac{S'}{\text{scale}_s}, V = \frac{V'}{\text{scale}_v} \quad (14)$$

$$\alpha = V \times (1 - S) \quad (15)$$

$$\beta = \begin{cases} \text{undefined} & \text{if } H \text{is undefined} \\ V \times (1 - (H - \lfloor H \rfloor) \times S) & \text{otherwise} \end{cases} \quad (16)$$

$$\gamma = \begin{cases} \text{undefined} & \text{if } H \text{is undefined} \\ V \times (1 - (1 - (H - \lfloor H \rfloor)) \times S) & \text{otherwise} \end{cases} \quad (17)$$

by https://mattlockyer.github.io/iat455/documents/rgb-hsv.pdf

Converting HSV to RGB

$$(R, G, B) = \begin{cases} (V, V, V) & \text{if } H \text{ is undefined} \\ (V, \gamma, \alpha) & \text{if } 0 \leq H < 1 \\ (\beta, V, \alpha) & \text{if } 1 \leq H < 2 \\ (\alpha, V, \gamma) & \text{if } 2 \leq H < 3 \\ (\alpha, \beta, V) & \text{if } 3 \leq H < 4 \\ (\gamma, \alpha, V) & \text{if } 4 \leq H < 5 \\ (V, \alpha, \beta) & \text{if } 5 \leq H < 6 \end{cases} \tag{18}$$

$$(R', G', B') = (R \times scale_r, G \times \text{scale}_g, B \times \text{scale}_b) \tag{19}$$

by https://mattlockyer.github.io/iat455/documents/rgb-hsv.pdf

Now, let's see how it actually works in practice!

Useful links

1. Supervisely for data labeling and annotation - https://app.supervise.ly/.
2. About OpenCV Library - https://opencv.org/.
3. OpenCV and colour models for image segmentation - https://realpython.com/python-opencv-color-spaces/.
4. Kaggle solutions for fresh and rotten fruits classification - https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification/notebooks.
5. Kaggle notebook showing how ordinary CNN model looks - https://www.kaggle.com/salmaachour/classification-cnn-vgg16.

Thank you for your attention!

Contacts:

Mail: Nikita.Stasenko@skoltech.ru

Telegram:@Stanix94