University of Sargodha  


Week 1 Work

Submitted By: -

Adeel Shahid.

Submitted To: -

Sir Fahad Maqbool.

Class: -

BSCS 7th (Self).

Roll no.: -

BSCSF17E045.

**Delta e algorithm:**

CIELAB ΔE\*[[edit](https://en.wikipedia.org/w/index.php?title=Color_difference&action=edit&section=4)]

The [International Commission on Illumination](https://en.wikipedia.org/wiki/International_Commission_on_Illumination) (CIE) calls their distance metric Δ*E*\**ab* (also called Δ*E\**, or, inaccurately, *dE*\*, *dE*, or "Delta E") where [delta](https://en.wikipedia.org/wiki/Delta_(letter)) is a [Greek letter](https://en.wikipedia.org/wiki/Greek_letter) often used to denote difference, and **E** stands for *Empfindung*; German for "sensation". Use of this term can be traced back to [Hermann von Helmholtz](https://en.wikipedia.org/wiki/Hermann_von_Helmholtz) and [Ewald Hering](https://en.wikipedia.org/wiki/Ewald_Hering).

Perceptual non-uniformities in the underlying [CIELAB](https://en.wikipedia.org/wiki/CIELAB) color space have led to the CIE refining their definition over the years, leading to the superior (as recommended by the CIE) 1994 and 2000 formulas.[[6]](https://en.wikipedia.org/wiki/Color_difference#cite_note-6) These non-uniformities are important because [the human eye is more sensitive to certain colors than others](https://en.wikipedia.org/wiki/Color_vision#Physiology_of_color_perception). CIELAB metric is used to define color tolerance of CMYK solids. A good metric should take this into account in order for the notion of a "[just noticeable difference](https://en.wikipedia.org/wiki/Just_noticeable_difference)" to have meaning. Otherwise, a certain Δ*E* may be insignificant between two colors in one part of the color space while being significant in some other part.[[7]](https://en.wikipedia.org/wiki/Color_difference#cite_note-7)

**CIE76**

The 1976 formula is the first formula that related a measured color difference to a known set of CIELAB coordinates. This formula has been succeeded by the 1994 and 2000 formulas because the CIELAB space turned out to be not as perceptually uniform as intended, especially in the saturated regions. This means that this formula rates these colors too highly as opposed to other colors.

Given two colors in [CIELAB color space](https://en.wikipedia.org/wiki/CIELAB_color_space), {\displaystyle ({L\_{1}^{\*}},{a\_{1}^{\*}},{b\_{1}^{\*}})} and {\displaystyle ({L\_{2}^{\*}},{a\_{2}^{\*}},{b\_{2}^{\*}})}, the CIE76 color difference formula is defined as:

{\displaystyle \Delta E\_{ab}^{\*}\approx 2.3}

**Otsu algorithm:**

In [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [image processing](https://en.wikipedia.org/wiki/Image_processing), **Otsu's method**, named after [Nobuyuki Otsu](https://en.wikipedia.org/wiki/Nobuyuki_Otsu) is used to perform automatic image [thresholding](https://en.wikipedia.org/wiki/Thresholding_(image_processing)).[[1]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-Mehmet-1) In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance.[[2]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-Otsu-2) Otsu's method is a one-dimensional discrete analog of [Fisher's Discriminant Analysis](https://en.wikipedia.org/wiki/Linear_discriminant_analysis#Fisher's_linear_discriminant), is related to [Jenks optimization method](https://en.wikipedia.org/wiki/Jenks_optimization_method), and is equivalent to a globally optimal [k-means](https://en.wikipedia.org/wiki/K-means_clustering)[[3]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-3) performed on the intensity histogram. The extension to multi-level thresholding was described in the original paper,[[2]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-Otsu-2) and computationally efficient implementations have since been proposed.[[4]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Otsu%27s_method#cite_note-5)

**Otsu algorithm:**

**function** level = otsu(histogramCounts)

total = sum(histogramCounts); *% total number of pixels in the image*

*%% OTSU automatic thresholding*

top = 256;

sumB = 0;

wB = 0;

maximum = 0.0;

sum1 = dot(0:top-1, histogramCounts);

for ii = 1:top

wF = total - wB;

if wB > 0 && wF > 0

mF = (sum1 - sumB) / wF;

val = wB \* wF \* ((sumB / wB) - mF) \* ((sumB / wB) - mF);

**if** ( val >= maximum )

level = ii;

maximum = val;

end

end

wB = wB + histogramCounts(ii);

sumB = sumB + (ii-1) \* histogramCounts(ii);

end

end

**k-mean cluster algorithm:**

# Modules

import matplotlib.pyplot as plt

from matplotlib.image import imread

import pandas as pd

import seaborn as sns

from sklearn.datasets.samples\_generator import (make\_blobs,

make\_circles,

make\_moons)

from sklearn.cluster import KMeans, SpectralClustering

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import silhouette\_samples, silhouette\_score

%matplotlib inline

sns.set\_context('notebook')

plt.style.use('fivethirtyeight')

from warnings import filterwarnings

filterwarnings('ignore')

# Import the data

df = pd.read\_csv('../data/old\_faithful.csv')

# Plot the data

plt.figure(figsize=(6, 6))

plt.scatter(df.iloc[:, 0], df.iloc[:, 1])

plt.xlabel('Eruption time in mins')

plt.ylabel('Waiting time to next eruption')

plt.title('Visualization of raw data');

**sine cosine algorithm***:*

The **Sine Cosine Algorithm (SCA)** is a new optimization technique for solving optimization problems. The SCA creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution using a mathematical model based on sine and cosine functions. Several random and adaptive variables also are integrated to this algorithm to emphasize exploration and exploitation of the search space in different milestones of optimization.

**Different type of symptoms and diseases Citrus Fruits:**

**Disease:**

***Fire Blight:***

This bacterial disease affect shoots wilt and look blackened. Effected plats are apples, pears, roses, small fruits and fruit trees.

**Picture:**



**Disease:**

***Anthracnose:***

Anthracnose is a fungal disease that tends to attack plants in the spring when the weather is cool and wet, primarily on leaves and twigs. The fungi overwinter in dead twigs and fallen leaves. Cool, rainy weather creates perfect conditions for the spores to spread

**Picture:**



**Disease:**

***Alternaria Blight (Early Blight):***

This fungal disease affects fruit trees, vegetables and shade trees. It effects on leaves, green to brown, brown to black spots form and enlarge, developing concentric rings on leaves.

**Picture:**



**Disease:**

***Canker:***

This bacterial disease affects main stem, branch or root. Woody stems, sunken areas, abnormal tissue, raised parts of dead typically form of Canker.

**Picture:**

****

**Disease:**

***Nectria Canker:***

Nectria Canker is a common fungus plant disease that attacks most vines, hardwoods and shrubs. It is maximum harmful in Maple trees. Sunken affected area appear small on the spore-producing structures and bark near wounds are formed in small pink color. It kills branches and twigs and may girdle young trees.

**Picture:**

****

**Disease:**

***Downy Mildew:***

Downy mildew is fungal disease that infect many vegetables, fruits, grasses and flowers. In older leaves this disease appears yellow to white region on the upper surfaces of the leaves. On the undersides, these leave areas are converted white to purple, growth frosts usually on the stems and underside of leaves which turn black with age.

**Picture:**

****

**Disease:**

***Crown gall:***

Crown gall is a common soil-borne bacterium plant disease that is found raspberries, grapes, roses and stone fruits. It also found herbaceous and woody shrubs plants.

**Picture:**

****

**Conclusion:**

*We will work on 3 diseases like Citrus Greening, Citrus Canker and Anthracnose, the algorithms we will use in our project are Delta E, L\*A\*b\*, k-mean cluster algorithm and sine cosine algorithm.*