

## K – Means Yöntemi İle Renkli Resim Bölütleme

Kenan POLAT 12011037

### 1. Yöntem

- Öncelikle renkli resmi okumak, yazmak ve resim değerlerine ulaşmak için openCV programından yararlanıldı.

Kullanılan OpenCV komutları

```
img=cvLoadImage(argv[1]);
```

Resmi kullanıcıdan alma

```
height    = img->height;
width     = img->width;
channels   = img->nChannels;
data      = (uchar *)img->imageData;
```

Resmin gösterilmesi;

```
cvShowImage("mainWin", img );
```

- Daha sonra RGB boyutunda olan resim uzayı HSV boyutuna dönüştürüldü.

```
void RGBtoHSV( float b, float g, float r, float *v, float *s, float *h )
```

- Kullanılan dönüşüm formülleri;

$$h = \begin{cases} 0 & \text{if } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 0^\circ) \bmod 360^\circ, & \text{if } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$$v = \max$$

- Daha sonra resmi bölütlemek için Hue değerleri alındı.
- K-Means yöntemini uygulamak için yazılan fonksiyon;

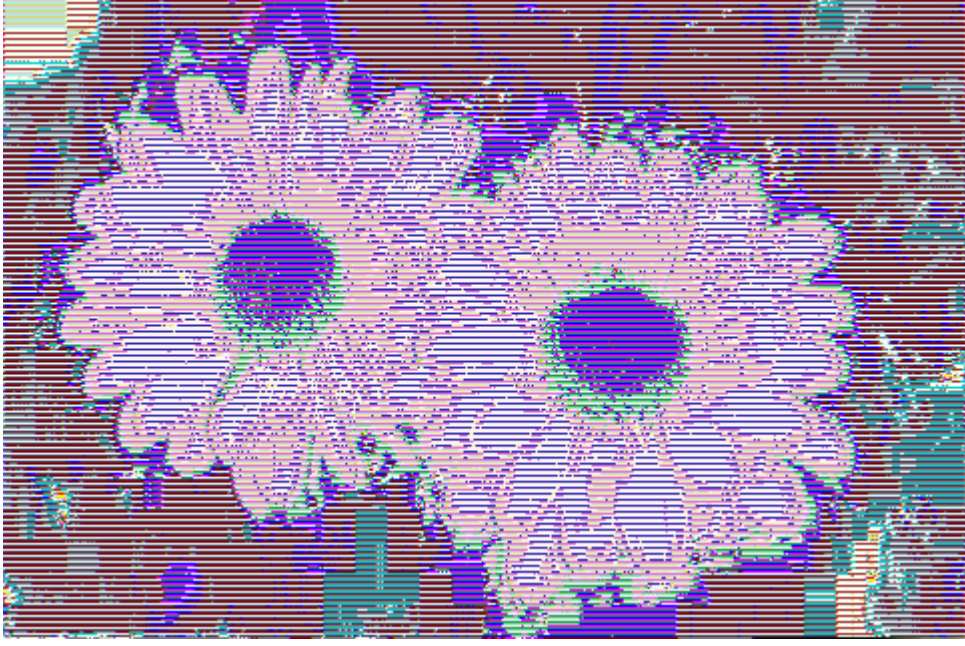
```
void k_mean_clustering( unsigned int *hue, int height, int width)
```

- Buradaki hue dizisi resimden elde edilen H değerleridir.
- Daha sonra pikseller arasında kümeleme yapmak için kullanıcıdan k değeri alındı. Rastgele seçilen 0- 360 arasında k tane değer `clusters[k]` dizisine atıldı. Daha sonra bütün pikseller taranarak kümeleme yaptığımız değerlere yakın pikseller aynı etiket ile etiketlendi.
- Etiketleme değerleri `tags` dizisinde tutuldu. Oluşturulan `tags` dizisi Connecting Component Labeling yapmak için `void labeling( int *tags, int height, int width)` fonksiyonuna gönderildi. Bu fonksiyonda da aynı etiketli komşuların bir bölüt oluşturması için her pikselin 4 komşusuna ve komşu etiketlerine bakıldı. Eşleşen komşuların etiket değerleri alındı. Eşleşmeyen piksellere yeni bölüt bölüt numarası verildi.
- Son kısımda ise `int getColors(int index,int rgb)` fonksiyonu ile oluşturulan renk matrisinden RGB uzayında renkler alınarak resmimizin bölütleri renklendirildi.

## 2. Uygulama



*K = 16 için Orijinal resim*

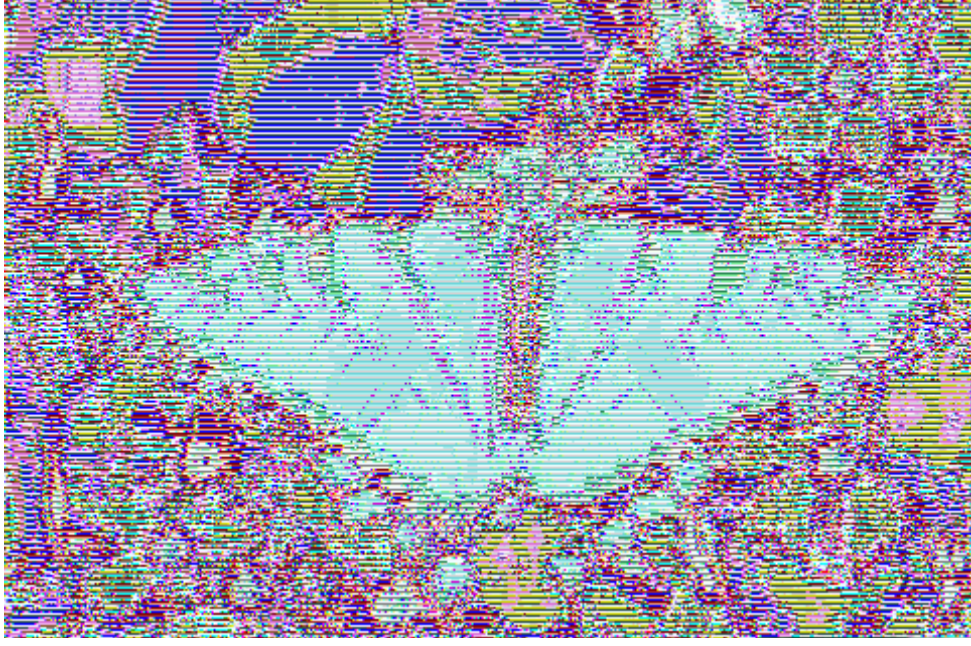


*K = 16 için Bölütlenmiş resim*



*K = 32 için Orjinal resim*

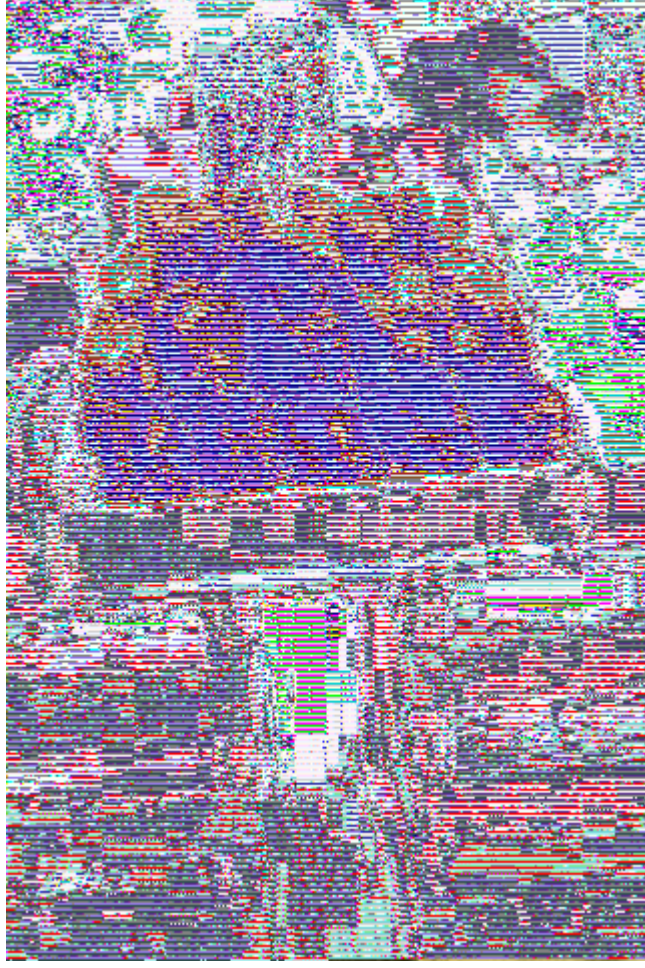




$K = 32$  için Bölütlenmiş resim



$K = 64$  için Orjinal resim



*K = 64 için Bölütlenmiş resim*

### 3. Sonuç

Özellikle k-means clustering yöntemi ile etiketleme işleminde k değeri arttıkça resimdeki piksellerin birbirinden daha çok ayrıldığını ve daha fazla bölütün oluştuğunu görüyoruz. Programın çalışmasını etkileyen kısımlardan bir diğeri ise kümelerin başlangıç ortalama değerlerinin random seçilme olayıdır. İlklenirmede birbirine yakın küme ortalamaları geldiği zaman, kümelemenin bitmesi için geçen adım süresi artmaktadır.