# Color Clustering

BLM 5110 Makine Öğrenmesi

Metin Uslu

235B7014

Yapılan çalışmanın kodlarına ColorClustering{.ipynb, .py ve .html} dosyaları üzerinden erişebilirsiniz. Projenin tüm repository’sine aşağıdaki adres üzerinden erişim sağlayabilirsiniz.

Github Color Clustering Repo: <https://github.com/metinuslu/blm5110_color_clustering>

**Ödev Raporu:** Ödev raporunuzu aşağıdaki sıra ile makale formatında yazınız.

1. Özet: Yaptığınız çalışmayı, elde ettiğiniz sonuçları özet olarak veriniz.

Bu çalışmada Roboflow platformundan alınan farklı çözünürlükte ve doğal ortamlarda çekilmiş, kırmızı, yeşil, mavi, beyaz ve gri renk çeşitliliğinde ürün görselleri Renk kanalları üzerinden kümelenerek renkler bazında sınıflandırılması amaçlanmıştır. Öncesinde resimlerin array haline getirilmesi, ardından renk kanallarının orijinal ve normalize histogramlarının elde edilmesi ve görselleştirilmesi sağlanmıştır. Devamında kendi yazdığım K-Means algoritması kullanılarak her resim için R, G ve B normalize histogramları üzerinden kümele çalışması yapılmıştır. **Sonuçlardan bahset.**

1. Giriş: Ödev konusunu tanıtan 1 paragraflık bir giriş yapınız. Bu çalışmanın nerelerde kullanılabileceğinden bahsediniz.

Bu çalışma da Ürün görsellerinin R,G,B Renk kanalları/özellikleri kullanılarak benzer renklerin kümelenmesi için yapılan bir çalışmasıdır. Bu çalışma;

1. Ürün görselleri üzerinde Renk özelliklerinin tespit edilmesinde kullanabilir.
2. Yine aynı zamanda ürünlerin renklerini kullanarak birbirlerine yakın olanları tespit ederek bunlar gruplanabilir.
3. Bu aynı zamanda sonsuz sayıda bulunan ürün renk uzayını azaltmaya da (reduce) de katkı sağlar. Modelleme açısından “High Cardinality” problemine de çözüm sağlayacaktır.
4. Sistem Tasarımı: Sisteminizin işlem adımlarını kısaca anlatınız.

**Sistem Tasarımında bahset.**

1. Resimlerin formatını hazır kütüphaneler kullanarak çözerek görüntüye ait matrisi elde ediniz. Bir görüntü piksellerin (R,G,B) bileşenlerinden oluştuğu bir matristir.

**create\_img\_to\_arr** fonksiyonu kullanılarak her bir resim dosyası array haline getirilmiştir. Burada resim dosyaları okunurken OpenCv library’si kullanılmıştır. OpenCv ile resimler varsayılan olarak BGR olarak okunmaktadır. Buna dikkat edilerek resim nesneleri BGR to RGB çevrilmiştir. Fonksiyon hem RGB hem de BGR resim arrayleri return etmektedir.

A screenshot of a computer

Description automatically generated

1. Resimleri oluşturan piksellerin (R,G,B) değerlerine göre her resmin renk histogramını (Her resimde R, G, B bileşenleri için ayrı ayrı olmak üzere toplam 3 histogram) elde ediniz. Histogram eldesi için de hazır kütüphane kullanabilirsiniz. Bir pikselin R,G,B bileşenlerinin değeri 0-255 arası değiştiği için her histogram dizisi 256 elemanlı olmalıdır.

**img\_to\_hist\_arr** fonksiyonu kullanılarak her bir resim nesnesinin R, G ve B histogramları OpenCv library’si kullanılarak elde edilmiştir. Fonksiyon R, G ve B için ayrı ayrı histogram arrayleri return etmektedir.

A screenshot of a computer

Description automatically generated

R, G ve B Histogramlarının görselleştirilmesi için **plot\_histogram** fonksiyonu yazılmıştır. Örnek birkaç resim için çıktıları aşağıda paylaşıyorum. ColorClustering Notebook (.ipynb, .html) üzerinden tamamına erişebilirsiniz.

A screenshot of a computer program

Description automatically generated

A group of graphs showing different colors

Description automatically generated

1. Her resimde, her renk bileşeni için histogram dizisindeki sonuçları resimdeki toplam piksel sayısına bölerek dizi elemanlarının değerini [0-1] aralığına normalize ediniz.

**img\_to\_norm\_hist\_arr** fonksiyonu kullanılarak R, G ve B histogramları 0 - 255 aralığından 0 - 1 aralığına normalize edilmiştir.

A screenshot of a computer

Description automatically generated

R, G ve B Histogram ve Normalize Histogram çıktılarını bir örnek resim üzerinden paylaşıyorum.

A screenshot of a computer screen

Description automatically generated

1. Resimlerin histogramlarını benzerliklerine göre k=5 değeri için(5 renk sınıfı olduğu için) k-means yöntemiyle kümeleyiniz. K-means’de başlangıç adımında rasgele seçeceğiniz k resmin histogramını k cluster’ın başlangıç merkezi olarak kullanınız.

KMeans class içerisine **fit**, **predict** ve centroid değerlerinin rastgele belirlenmesi için **initialize\_centroids** fonksiyonları yazılmıştır.

A screenshot of a computer

Description automatically generated

1. Deneysel Sonuçlar: Sistem başarısını değerlendirmek için aşağıdaki işlemleri yapınız.
   1. Küme merkezlerini rasgele belirleyerek her k değeri için kümeleme işlemini 10 defa tekrarlayınız. Her kümeleme sonunda, kümelerde doğru cluster’da olan resim yüzdesini hesaplayınız.
   2. Hazırlayacağınız karışıklık matrisi (confusion matrix) üzerinde sonucu gösteriniz.
   3. Her sınıf için doğru cluster’a yerleşmiş 5 örnek ve yanlış cluster’da bulunan 1 örnek resmi veriniz.
2. Sonuç: Sizce kümeleme işlemi başarılı oldu mu? Yanlış kümelerde olan resimler sizce neden yanlış kümelerde yer aldı? Sistemin genel başarısını yorumlayınız. Başarısızlık sebebi olduğunu düşündüğünüz problemlerin giderilmesi için varsa önerilerinizi belirtiniz.