Görüntülerden Hava Durumu Kümeleme Problemi

180201109 - Poyraz Aktaş – [180201109@kocaeli.edu.tr](mailto:180201109@kocaeli.edu.tr)

180201026 – Hamza Tunahan Arslan – [180201026@kocaeli.edu.tr](mailto:180201026@kocaeli.edu.tr)

180201045 - Orkun Alkan – [180201045@kocaeli.edu.tr](mailto:180201045@kocaeli.edu.tr)

180201057 - Oğuz Can Varol – [180201057@kocaeli.edu.tr](mailto:180201109@kocaeli.edu.tr)

# Özet

Görüntülerden hava durumu kümeleme problemi, hareket halindeki otonom arabalardan alınan görüntülerin aşağıdaki beş hava durumu kategorisinden birinde sınıfladırılmasıdır.

Beş hava durumu kategorisi aşağıdaki gibidir:

1. Hava açık
2. Yol biraz ıslak ve hava biraz bulutlu
3. Islak yol, bulutlu hava ve hafif yağmur
4. Alacakaranlıkta veya şafak vaktinde yağmurlu hava.
5. Gece şiddetli yağmur ve ıslak yollar.

Projenin açıklamasını içeren web sitesindeki veri setini bu beş hava durumu kategorisinde kümelemek için otonom arabalardan alınan görüntülerin özelliklerini Keras framework’ünün InceptionV3, VGG16 ve MobileNet gibi görüntü sınıflandırma modelleriyle çıkarttık. Çıkardığımız bu özelliklere göre görüntüleri beş ayrı kümeye ayırmak için *k-Means* algoritmasını kullandık.

K-Means algoritmasının bu görüntü sınıflandırma modellerinin hangisiyle daha iyi çalıştığını ölçmek için Silhoutte Score ve Calinski-Harabasz Index performans metriklerini kullandık.

Sonuç olarak MobileNet görüntü sınıflandırma modelinin bu 5 kümeyi daha iyi ayırdığı kanısına vardık.

# Giriş

Görüntülerden hava durumu kümeleme problemi, hareket halindeki otonom arabalardan alınan görüntülerin beş hava durumu kategorisinden birinde sınıfladırılmasıdır.

Problemi çözerken benzer problemler ve çözümleri hakkında bir literatür taraması yaptık. Yaptığımız bu taramada [kaggle](https://www.kaggle.com/code/prakharrathi25/weather-data-clustering-using-k-means/notebook) üzerinde günlük hava durumu verilerinin k-Means algoritmasıyla soğuk, sıcak ve kuru hava olacak şekilde üç kümeye ayrılmasını sağlayan bir çalışmayı ve [towardsdatascience](https://towardsdatascience.com/image-clustering-using-k-means-4a78478d2b83) üzerinde bulunan bir blogun InceptionV3 image classification modeli ve KMeans algoritmasını kullanarak kedi ve köpek kümelemesini yapan bir çalışmayı örnek aldık.

Sınıflandırma problemlerinde etiketli veri ya da eğitim verisi olmadığında bu problemi çözmek için benzer görüntüleri kümeleyebiliriz.

Kümeleme(*Clustering*) benzer özellikteki verileri grupladığımız denetimsiz bir makine öğrenmesidir. Bunu girdileri yorumlayıp özellik uzayından doğal -kümeler- gruplar oluşturarak yapar.

Kümeleme algoritmalarının ne kadar başarılı çalıştığını ölçmek için bazı performans metrikleri vardır. Bu performans metrikleri genel olarak küme içi(*intra-cluster*) uzaklıların en az ve kümeler arası(inter-cluster) uzaklıkların en fazla olmasına göre skorlar oluşturur. Biz de bu performans metriklerinden bizim geliştirdiğimiz modele uygun olan iki tanesi ile modelimizin performansını ölçtük.

# Yöntem

Sınıflandırma ve Regresyon görevleri Denetimli Öğrenme denilen şeyi oluştururken, Kümeleme Denetimsiz Öğrenme görevlerinin çoğunu oluşturur. Bu iki makro alan arasındaki fark, kullanılan veri türünde yatmaktadır. Denetimli Öğrenmede örnekler kategorik bir etiketle (Sınıflandırma) veya sayısal bir değerle (Regresyon) etiketlenirken, Denetimsiz Öğrenmede örnekler etiketlenmez, bu da gerçekleştirmeyi ve değerlendirmeyi nispeten karmaşık bir görev haline getirir.

## Image Classification Models in Keras

### VGG16

Çok fazla hyperparam yerine daha sade bir yapı kullanılır

Sinir ağları mimarisini **sadeleştirir**

16 ifadesi parametreli 16 katmanı olduğu anlamına gelir

138m parametresi vardır bu da normale göre oldukça fazladır.

### Inceptionv3

Inception v3 Google'ın Inception Convolutional Neural Network'ün üçüncü baskısıdır

Görüntü analizine ve nesne algılamaya yardımcı olmak için oluşturulan bir sinir ağıdır.

Inceptionv3'ün tasarımı, daha derin ağlara izin verirken aynı zamanda parametre sayısının çok fazla büyümesini önlemeyi amaçlıyor.

### Mobilenet

Mobilenet, Normal CNN'ler tarafından yapılan normal evrişimden farklı olan Derinlik evrişim ve nokta evrişim fikrini kullanır. Bu evrişim yolları, karşılaştırma ve tanıma süresini çok azalttığından, çok kısa sürede daha iyi bir yanıt sağlar.

Ağlarda aynı derinliğe sahip düzenli kıvrımlara sahip ağ ile karşılaştırıldığında parametre sayısını önemli ölçüde azaltır ve bu, CNN'nin görüntüleri tahmin etme verimliliğini arttırır, dolayısıyla mobil sistemlerde de rekabet edebilirler ve oldukça sık kullanılırlar.

Düşük hesaplama tüketimi ve hızlı çalışma hızıyla cep telefonlarında çalışırlar, bu nedenle mobil terminallerdeki uygulamalar için çok uygundurlar.

| **Model** | **Size** | **Parameters** |
| --- | --- | --- |
| VGG16 | 528 MB | 138, 400,000 |
| Mobile Net | 16 MB | 4,300,000 |
| Inception v3 | 92 MB | 23,900,000 |

## Kümeleme Algoritmaları

### K-Means Kümeleme Nedir ?

K-means kümeleme (*K-means clustering*) yöntemi *N* adet veri nesnesinden oluşan bir veri kümesini giriş parametresi olarak verilen K adet kümeye bölümlemektir. Amaç, gerçekleştirilen bölümleme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin ise minimum olmasını sağlamaktır.

K-Means algoritması bir unsupervised learning(gözetimsiz öğrenme) ve kümeleme algoritmasıdır.

#### Unsupervised Learning Nedir?

Bu yöntemde işaretlenmemiş (unlabelled) veri üzerinden bilinmeyen bir yapıyı tahmin etmek için bir fonksiyon kullanan makine öğrenmesi tekniğidir. Burada girdi verisinin hangi sınıfa ait olduğu belirsizdir.

### K-Means Algoritması Nasıl Çalışır ?

K-means algoritmasının çalışma mekanizmasına göre öncelikle her kümenin merkez noktasını veya ortalamasını temsil etmek üzere K adet nesne rastgele seçilir. Kalan diğer nesneler, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak en benzer oldukları kümelere dahil edilir. Daha sonra, her bir kümenin ortalama değeri hesaplanarak yeni küme merkezleri belirlenir ve tekrar nesnelerin merkeze uzaklıkları incelenir. Herhangi bir değişim olmayıncaya kadar algoritma tekrarlamaya devam eder.

Algoritma temel olarak 4 aşamadan oluşur:

1.Küme merkezlerinin belirlenmesi

2.Merkez dışındaki verilerin mesafelerine göre kümelendirilmesi

3.Yapılan kümelendirmeye göre yeni merkezlerin belirlenmesi (veya eski merkezlerin yeni merkeze kaydırılması)

4.Kararlı hale (stable state) gelinene kadar 2. ve 3. adımların tekrarlanması.

## Performans Metrikleri

### Silhouette Score

Silhouette Score bir kümeleme tekniğinin iyiliğini hesaplamak için kullanılan bir ölçüdür. Değeri -1 ile 1 arasında değişir.

1: Kümelerin birbirinden oldukça uzak olduğu ve açıkça ayırt edildiği anlamına gelir.

0: Kümelerin kayıtsız olduğu anlamına gelir veya kümeler arasındaki mesafenin önemli olmadığını söyleyebiliriz.

-1: Kümelerin yanlış şekilde atandığı anlamına gelir.

#### Formül

Silhouette Score = (b-a)/max(a,b)

a = ortalama küme içi mesafe, yani bir küme içindeki her nokta arasındaki ortalama mesafe.

B = ortalama kümeler arası mesafe, yani tüm kümeler arasındaki ortalama mesafe.

### Calinski-Harabasz Index

Calinski-Harabasz İndeksi, Varyans Oranı Kriteri olarak da bilinir.

Puan, küme içi dağılım ile kümeler arası dağılım arasındaki oran olarak tanımlanır. C-H İndeksi, kesin doğruluk etiketleri hakkında bilgi gerektirmediğinden, Kümeleme algoritmasının performansını değerlendirmenin harika bir yoludur.

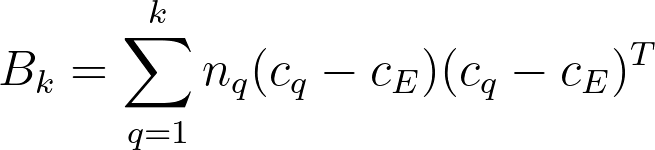
Index ne kadar yüksek olursa, performans da o kadar iyi olur.

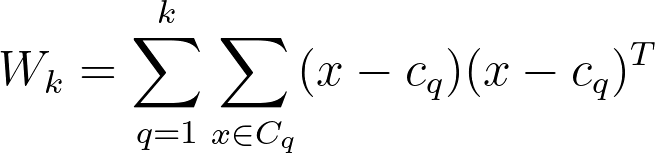
#### Calinski-Harabasz Index Formül

A picture containing text, blackboard

Description automatically generated

burada tr(Bk) grup arası dağılım matrisinin izidir ve tr(Wk) şu şekilde tanımlanan küme içi dağılım matrisinin izidir:





## Projede Kullanılan Teknolojiler

* Keras
* Tensorflow
* Python
* Jupyternotebook
* Vscode
* Miniconda

# Sonuç

Inceptionv3, VGG16 ve Mobilenet image classification modellerini karşılaştırdık. Bu modellerden çıkarttığımız özellik uzayını K-Means algoritmasında 5 küme oluşturarak aşağıdaki sonuçları elde ettik.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **VGG16** | **Inceptionv3** | **Mobilenet** |
| **Silhouetter** | -0.016077179461717606 | 0.017532600089907646 | 0.022400561720132828 |
| **Calinski Harabasz** | 11.388337858939199 | 11.42150423188575 | 11.649970680615386 |

Karşılaştırdığımız görüntü sınıflandırma modelleri arasında hem Silhoutte hem de Calinski Harabasz Index skorlarında en yüksek dereceye ulaşan Mobilenet modelinin bizim problemimiz için en uygun sonucu verdiğini göstermiş olduk.

# Kaynakça

<https://towardsdatascience.com/performance-metrics-in-machine-learning-part-3-clustering-d69550662dc6>

<https://keras.io/api/applications/#usage-examples-for-image-classification-models>

<https://ai.yemreak.com/deeplearning.ai/cnn/klasik-cnn-modellerini-inceleme>

<https://www.mygreatlearning.com/blog/introduction-to-vgg16/>

<https://arxiv.org/pdf/1704.04861.pdf>

<https://gutx.com.tr/item/arudevmf7o5q.html>

<https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>

<https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>

<https://www.geeksforgeeks.org/vgg-16-cnn-model/>

<https://medium.com/deep-learning-turkiye/k-means-algoritmas%C4%B1-b460620dd02a>

<https://erdincuzun.com/makine_ogrenmesi/hangisini-secmeliyim-supervised-ve-unsupervised-learning/>

<https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c>