**ALGORİTMA ANALİZİ VE TASARIMI DERSİ 2. KISA SINAV 2. ÖDEVİ-GÖRSEL TANIMA/TESPİT ETME**

<https://github.com/arzuv-hudaynazarova/Python_OpenCV_Comparison>

https://github.com/Joseph0grcn/image\_comprasion

**Ad:** Yusuf

**Soyadı:** Gürcan

**Öğrenci-No:** 1180505810

**Ad:** Arzuv

**Soyadı:** Hudaynazarova

**Öğrenci-No:** 5200505059

Birbirine benzeyen görselleri bulmak için genellikle derin öğrenme tabanlı yöntemler kullanılır. Bu yöntemler genellikle bir sinir ağı modelini kullanırlar, örneğin bir **Convolutional Neural Network (CNN),** görsellerin özelliklerini (features) çıkarır ve bu özellikler arasındaki benzerlikleri bulur.

Bunun yanında, **OpenCV** gibi kütüphaneler de görsel benzerlik bulma problemlerini çözmek için kullanılabilir**. *Template matching, histogram* *comparison*** ve ***feature matching*** gibi çeşitli yöntemler bulunmaktadırlar. Ancak, bu tür yöntemler genellikle basit veya düşük seviye görsel benzerliklerini bulmak için daha uygundur.

Özellikle, 20x20 piksel gibi küçük parçalar üzerinden benzerlik bulma durumunda, bir **deep learning** modelini eğitmek çözüm sağlayabilir. Ancak, bu durumda, modelin eğitilmesi için geniş bir veri setine ihtiyaç duyulacaktır.

Algoritmanın karmaşıklık sınırları genellikle seçilen yönteme ve kullanılan veri setinin büyüklüğüne bağlıdır. Derin öğrenme modelleri genellikle büyük miktarda hesaplama kaynağı gerektirir ve eğitim süreleri uzun olabilir. Ancak, bir kere eğitildikten sonra, yeni görseller üzerinde benzerlik bulma işlemi genellikle hızlıdır.

**OpenCV** gibi kütüphanelerle yapılan yöntemler genellikle daha az hesaplama kaynağı gerektirir, ancak daha basit veya düşük seviye benzerliklerin bulunmasında daha etkilidirler. Bu tür yöntemlerin karmaşıklığı genellikle **O(n^2)** veya daha kötü olabilir, burada **n**, işlenen görsel sayısıdır.

Bu nedenle, en uygun yöntemi seçerken, kullanılabilir kaynakları, gereksinimleri ve beklentileri dikkate almak önemlidir.

Görsel benzerliklerini belirlemek için birçok yöntem bulunmaktadır. Genellikle, bu tür bir problem bir tür görüntü analizi gerektirir. Görüntüler arasındaki benzerliği ölçmek için genellikle bir tür özellik çıkarımı (feature extraction) ve ardından bir özellik karşılaştırması (feature comparison) gerçekleştirilir.

Birbirine benzeyen görselleri bulmak için kullanılabilecek yöntemlerden bazıları:

OpenCV Image Template Matching: Bu yöntem, bir görüntü içinde belirli bir şablonu arar. Bu, özellikle belirli bir objeyi veya deseni aradığınızda işe yarar. Ancak, genellikle yalnızca bir görüntü içindeki eşleşmeleri bulmak için kullanılır, bu yüzden birçok farklı görüntü arasında genel benzerlikleri bulmak için ideal olmayabilir.

Özellik Tabanlı Yöntemler (Feature-Based Methods): Bu yöntemler genellikle bir görüntünün belirli özelliklerini (renk, doku, şekil vb.) çıkarır ve bu özellikleri karşılaştırır. Örneğin, SIFT, SURF, ORB gibi özellik çıkarıcılar ve bu özellikleri karşılaştırmak için kullanılan yöntemler (örneğin, Brute-Force matcher, FLANN tabanlı matcher vb.) bulunmaktadır.

Derin Öğrenme Tabanlı Yöntemler (Deep Learning-Based Methods): Derin öğrenme modeli, büyük bir veri setinde öğrenme yeteneğine sahip olabilir ve bu da onu çok farklı görüntüler arasındaki benzerlikleri bulmak için ideal hale getirebilir. Convolutional Neural Networks (CNN'ler) genellikle bu tür bir işlem için kullanılır. Özellikle, bir görüntüyü bir dizi özellik vektörüne dönüştüren bir CNN modeli kullanılabilir ve bu vektörler daha sonra benzerlik için karşılaştırılabilir.

Bu tür bir işlem genellikle karmaşıktır ve işlem süresi ve hafıza kullanımı, kullanılan yönteme, görüntülerin sayısına ve boyutlarına, ve belirli bir görev için gereken hassasiyete bağlıdır. Örneğin, derin öğrenme yöntemleri genellikle daha karmaşık ve işlem yoğundur, ancak daha genel ve güçlü sonuçlar verebilir.

Python dilinde, bu yöntemler kullanılarak iki görselin benzerlik oranı hesaplanabilir. Ancak, bu yöntemlerin tam bir görsel listesinde, özellikle de 20x20 piksel parçalar üzerinden çalıştırılması daha karmaşık bir işlem gerektirir. İki görselin karşılaştırılması durumunda, her iki görsel de aynı boyutta olmalıdır. Bu nedenle, 20x20 piksel parçaların benzerliklerini karşılaştırmak için, bu parçaların her birini ayrı ayrı karşılaştırmak gerekecektir.

Bu işlemin karmaşıklığı, karşılaştırılacak görsel sayısına ve her bir görselin boyutuna bağlıdır. Eğer N adet görsel ve her görselde M adet 20x20 piksel parça varsa, tüm parçaların karşılaştırılması için O(N^2 \* M^2) işlem gerekecektir. Bu, büyük görsel listeleri için oldukça zaman alıcı olabilir.

OpenCV'nin Template Matching yöntemi de bir başka seçenek olabilir. Ancak, bu yöntem genellikle bir görselde belirli bir şablonun varlığını tespit etmek için kullanılır, bu nedenle bu durumda doğrudan uygulanması zor olabilir.

Görseller üzerinde bu tür bir analiz yapmak için bir Python scripti oluşturmanız gerekecektir. Ancak, bu işlemi yaparken, özellikle büyük miktarda görsel üzerinde çalışıyorsanız, işlem süresinin ve karmaşıklığının artabileceğini göz önünde bulundurmanız önemlidir.