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

Ad: Ali Soyad:Altunar Numara:1210505032

Ad:Abdülkadir Soyad:Hocaoğlu Numara:1210505029

Ad:Muhammed Emin Soyad:Durmuş Numara:1210505049

Görseller arasındaki benzerlikleri bulmak için genellikle derin öğrenme tabanlı yöntemler tercih edilir. Bu yöntemler genellikle sinir ağı modelleri, örneğin Convolutional Neural Network (CNN), kullanır. CNN, görsellerin özelliklerini çıkarır ve bu özellikler arasındaki benzerlikleri bulmak için kullanılır.

Bunun yanı sıra, OpenCV gibi kütüphaneler de görsel benzerlik bulma problemlerini çözmek için kullanılabilir. Template matching, histogram karşılaştırma ve özellik eşleştirme gibi çeşitli yöntemler bulunmaktadır. Ancak, bu yöntemler genellikle daha basit veya düşük seviyeli görsel benzerliklerini bulmak için daha uygundur.

Özellikle 20x20 piksel gibi küçük parçalar üzerinde benzerlik bulmak için, bir derin öğrenme modeli eğitmek faydalı olabilir. Ancak, böyle bir durumda modelin eğitimi için geniş bir veri setine ihtiyaç duyulacaktır.

Yöntemin karmaşıklığı, seçilen yönteme ve kullanılan veri setinin büyüklüğüne bağlıdır. Derin öğrenme modelleri genellikle büyük miktarda hesaplama kaynağı ve uzun eğitim süreleri gerektirir. Ancak, bir kez eğitildikten sonra, yeni görseller üzerinde benzerlik bulma işlemi genellikle hızlı olur.

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

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 görüntü analizi gerektirir. Görseller arasındaki benzerliği ölçmek için genellikle bir tür özellik çıkarımı ve ardından bir özellik karşılaştırması gerçekleştirilir.

Birbirine benzeyen görselleri bulmak için kullanılabilecek yöntemlerden bazıları şunlardır:

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

Özellik Tabanlı Yöntemler: 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: Derin öğrenme modelleri, büyük bir veri setinde öğrenme yeteneğine sahip olabilir ve bu da onları çok farklı görseller arasındaki benzerlikleri bulmak için ideal hale getirir. 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. Derin öğrenme yöntemleri genellikle daha karmaşık ve işlem yoğundur, ancak daha genel ve güçlü sonuçlar verebilir.

Python dilinde, bu yöntemler kullanılarak iki görselin benzerlik oranını hesaplamak mümkündür. Ancak, bu yöntemlerin tam bir görsel listesi üzerinde, özellikle de 20x20 piksel parçalar üzerinde 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 bulunuyorsa, 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 seçenek olabilir. Ancak, bu yöntem genellikle bir görselde belirli bir şablonun varlığını tespit etmek için kullanılır, bu nedenle doğrudan bu duruma uygulanması zor olabilir.

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