

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



YAPAY ZEKA PROJE ÖDEVİ RAPORU

Öğrenci No: 19011038

Öğrenci Adı Soyadı: Şevval Bulburu

Öğrenci E-Posta: sevval.bulburu@std.yildiz.edu.tr

Öğrenci No: 20011023

Öğrenci Adı Soyadı: Mehmet Alperen Ölcer

Öğrenci E-Posta: alperen.olcer@std.yildiz.edu.tr

Video Linki: <https://youtu.be/orxbHXTbhis>

Ders/Grup: BLM4510 Yapay Zeka

Ders Yürütücüsü

PROF DR MEHMET FATİH AMASYALI

1 Haziran 2023

PROJENİN AMACI:

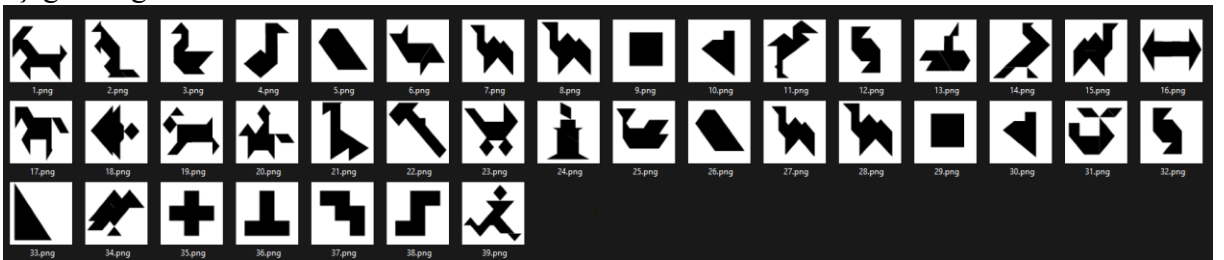
Bu projede tangram oyununa benzer bir yapı tasarlanmak istenmiştir. 9 adet farklı boyutlardaki dikdörtgenler kullanılarak girdi olarak verilen resimlerde; maksimum doluluk ve minimum taşma kriteri sağlanacak biçimde dikdörtgenler resmin içerisine yerleştirilmeye çalışılmıştır. Bu kriterleri sağlayarak dikdörtgenleri resimlerin içerisine yerleştirmek için de DFS mantığından yararlanarak çalışan bir arama fonksiyonu tasarlanmıştır.

DATASET HAZIRLANIŞI:

Öncelikle dataset hazırlamak için beyaz fon üzerine siyah resim olacak şekilde tangramlarda kullanılan şekilleri de barındıran 107 adet resim bulunmuştur. Oluşturulacak dataset için elde edilen resimler; formatları, boyutları, içerdikleri piksellerin siyah – beyaz oranları veya piksellerin RGB değerlerinin siyah ve beyaz dışında gri tonlarında renkleri içermesi gibi özelliklerinden dolayı aynı formatta bulunmamaktadır. Bu sebeple çeşitli eleme ve dönüştürme işlemlerinden geçirilmişlerdir.

- Öncelikle resimlerin bir kısmı jpeg formatında iken diğerlerinin png formatında olmaması için `convert_images_to_png()` fonksiyonu oluşturulmuştur. Bu fonksiyon ile girdi olarak verilen bütün resimler png formatına dönüştürülerek format farkı ortadan kaldırılmıştır.
- Algoritma tasarımlarında kullanılacak resimler içerisinde sadece siyah ve beyaz renkleri barındırmalıdır. Elde edilen resimlerde gri, beyaz ve siyah tonları olması bu duruma ters düşmektedir. Farklı içerikteki resimlerin elenmesi için `calculate_image_info()` fonksiyonu oluşturulmuştur. Bu fonksiyon ile her bir resim içerisindeki siyah ve beyaz pikseller sayılarak toplamlarının toplam piksel sayısına eşitliği kontrol edilerek sadece siyah ve beyaz renkleri içeren resimler elde edilmiştir.
- Girdi olarak verilen resimlerdeki piksel sayıları aynı olmalıdır. Bunun için işlemlerden geçmiş olan bütün resimlerin boyutlarını aynı yapmak için `resize_images()` fonksiyonu oluşturulmuştur. Bütün resimler 500*500 olmak üzere 25000 adet piksel içerecek şekilde yeniden düzenlenmiştir.
- Son olarak verilen resimlerin içerisindeki şekillerin ortalama olarak aynı boyutta olması beklenmektedir. Bunu sağlamak için `save_images_within_ratio()` fonksiyonu yazılmıştır. Bu fonksiyonda siyah piksellerin toplam piksel sayısına oranı 0.2 – 0.3 aralığında olan resimler kabul edilmiş, kalan resimler elenmiştir.

İşlemler sonucunda 39 adet resim ile algoritma çalıştırılmıştır. Elde edilen dataset aşağıdaki gibidir:



ALGORİTMA

Algoritmada kullanılan önemli fonksiyonların açıklamaları:

- `create_tangram_elements_list`: Fotoğrafların siyah piksel oranı %20-%30 olacak şekilde filtrelemiştik. Bu fonksiyonda ise input verilen fotoğrafın siyah piksel oranına göre farklı büyüklüklerde dikdörtgen listesi döndürülmektedir. Dikdörtgenlerin toplam alanı fotoğrafların siyah piksel oranı ile doğru orantılıdır.
- `combine_images`: Parametre olarak verilen büyük binary image'ın yine parametre olarak verilen koordinatına ufak binary image'ı yerleştirir ve OR operasyonu uygular. Bu işlem tangram elementinin tahtaya yerleştirme denemesinde etkisini gerçekleştirir.
- `find_coordinates_of_possible_combines`: Verilen fotoğrafta verilen boyutlarda dikdörtgen kaydırarak durumların içerdiği siyah piksel sayılarını saklar. Sonrasında Kmeans algoritması yardımıyla kümeleme yapıp birbirinden uzak ve en çok siyah yer kaplayacak koordinatları bulur.
- `find_outlier`: Yukarıda bulunan koordinatların işleme girme sırasını belirlerken en ayırık olanı önce takip edilir arama ağacında. Bunun için bu fonksiyondan yardım alınır. Kendisine verilen x, y koordinatlarından en ayırık olanı döndürür.
- `check_the_end_condition`: DFS'te durma koşulu bu fonksiyona bağlıdır. Maksimum doluluk ve minimum taşma kriterine göre karar verir.

DFS'te hızlı sonuç almak adına yapılan iyileştirmeler:

- Dikdörtgen yerleştirme sırası büyükten küçüğe olacak şekilde düzenlendi.
- Dikdörtgenin nereye yerleşeceği konusunda atlamalı kaydırma yapılarak 10000 kere kontrol işlemi olacak şekilde düzenlendi.
- 10000 olasılıktan 1000 en iyi olasılık seçilerek işlemlere öyle devam edildi.
- Kmeans algoritması yardımı ile kümeleme yapılarak yan yana iyi sonuçlardan kaçınıldı ve potansiyel konum 5'e düşürüldü.
- Bu 5 noktadan ise en ayırık olana öncelik verilip ağaçta ilerleyişte onunla devam edildi.

Sayısal başarı ölçütü:

- Başarı ölçütü verilen ilk tangram haritasında %70'den fazla doluluk ve taşmaların %5'in altında olmasıdır.
- Bir başka ölçüt ise çözüm bulmada gerçekleşen adım sayısıdır.

Yorumlar:

- Çözüm ağacın en altında olduğundan ötürü BFS araması kullanılmış olsaydı hafıza sorunları yaşanabilirdi.
- Çözüm ağacın en altında olduğundan ötürü DFS ile arama yapmak da işlem yükü açısından zorlayıcıydı. Olabildiğince doğru yolları tercih etmeye yönelik geliştirmeler yapıldı. Buna rağmen bazen çözüm bulmak çok uzun sürebiliyor.

ÇIKTILAR:

