



Vize Raporu

Eskişehir Osmangazi Üniversitesi

Image Processing

152117005

Özlem Kayıkcı

152120191043

Prof. Dr. Kemal ÖZKAN

2022-2023

İçindekiler

AMAÇ.....	3
KULLANILAN TEKNOLOJİLER	3
? Numpy kütüphanesi,	3
? DTW kütüphanesi,	3
? Matplotlib kütüphanesi,	3
? Colab (Google Colaboratory),	3
DTW (Dynamic Time Warping)	3
KODLAR VE VERİLERİN KARŞILAŞTIRMASI	4
1.Normal ve iki kat hızlıdaki video verisetinin karşılaştırılması	4
2.Öğretmen ve öğrencileri video verisetinin karşılaştırılması	5

AMAÇ

Verilen verisetlerinde örnek verisetlerinin referans verisetine göre hizalanması, eşleştirilmesi ve görselleştirilmesi suretiyle iki veriseti arasındaki benzerliğin bulunması ve videolar hakkında çıkarımlar yapılması amaçlanmıştır.

KULLANILAN TEKNOLOJİLER

- **Numpy kütüphanesi**, Python'da sayısal hesaplama yapmak için kullanılan bir kütüphanedir. Zaman serilerinin hesaplanması ve DTW algoritması için gerekli olan matris işlemleri için numpy kütüphanesi kullanılır.
- **DTW kütüphanesi**, Python'da DTW algoritmasını uygulamak için kullanılan bir kütüphanedir. Bu kütüphane, iki zaman serisi arasındaki benzerliği hesaplamak için kullanılan DTW algoritması için gerekli fonksiyonları içerir.
- **Matplotlib kütüphanesi**, Python'da görselleştirme işlemleri yapmak için kullanılan bir kütüphanedir. DTW algoritması sonucu elde edilen benzerlik matrisi görselleştirilerek, iki zaman serisi arasındaki benzerlik grafiği çizilir.
- **Colab (Google Colaboratory)**, Google tarafından sunulan, bulut tabanlı bir Jupyter defteri ortamıdır. Colab, Python programlama dilindeki kodların, tarayıcı üzerinden kullanılabilen bir arayüzde yazılmasına, çalıştırılmasına ve paylaşılmasına olanak tanır. Colab, kullanıcıların kendi bilgisayarlarında herhangi bir kurulum gerektirmeden Python kodlarını yazıp çalıştırmalarını sağlar. Colab, Google Drive'a kaydedilen Jupyter defterlerinin açılmasına, düzenlenmesine ve paylaşılmasına izin verir.

DTW (Dynamic Time Warping)

Dynamic Time Warping algoritması, iki farklı zaman serisinin benzerliğini ölçmek için kullanılan bir yöntemdir. Bu yöntem, iki zaman serisinin birbirleriyle eşleştirilmesi işlemi sırasında, zaman eksenini boyunca birbirine benzeyen noktaların eşleştirilmesi ile benzerlik ölçümü yapar. DTW'nin temel amacı, iki farklı zaman serisi arasındaki benzerliklerin ve farklılıkların ölçülmesidir. Bu yöntem, özellikle konuşma tanıma, müzik tanıma, el yazısı tanıma gibi alanlarda kullanılır. DTW, zaman serilerinin farklı uzunlukta olabileceği durumlarda da etkili bir şekilde çalışabilir, çünkü zaman serilerini dinamik olarak eşleştirir ve böylece uzunluk farklarına uyum sağlayabilir. Bu uygulamada DTW algoritmasından yararlanılmıştır.

KODLAR VE VERİLERİN KARŞILAŞTIRMASI

1.Normal ve iki kat hızdaki video verisetinin karşılaştırılması

```
import numpy as np
!pip install dtw
from dtw import dtw
import matplotlib.pyplot as plt

# İlk veri kümesi
ref_data1 = np.array([33.57, 3.25, 33.49, 128.64, 141.65, 141.13, 154.88, 123.91, 134.82, 133.98,
137.43, 140.52, 142.58, 41.79, 14.85, 38.24, 138.94, 134.13, 161.86, 167.1,
148.82, 23.72, 24.91, 69.76, 68.65, 136.73, 174.4, 176.15, 174.05, 174.6,
172.25, 171.85, 164.98, 147.41, 90.51, 38.16, 35.29, 22.73, 27.65, 147.68,
158.48, 86.7, 94.16, 39.58, 49.7, 31.96, 68.1, 138.98, 134.18, 177.62, 177.49,
167.42, 173.83, 173.18, 170.41, 164.65, 148.96, 58.94, 30.08, 24.64, 26.86])

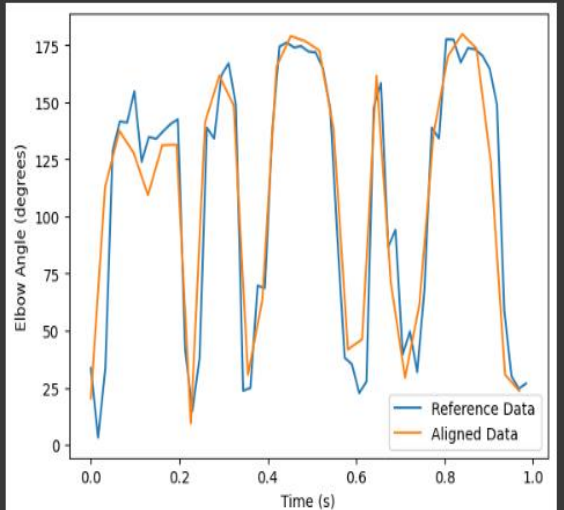
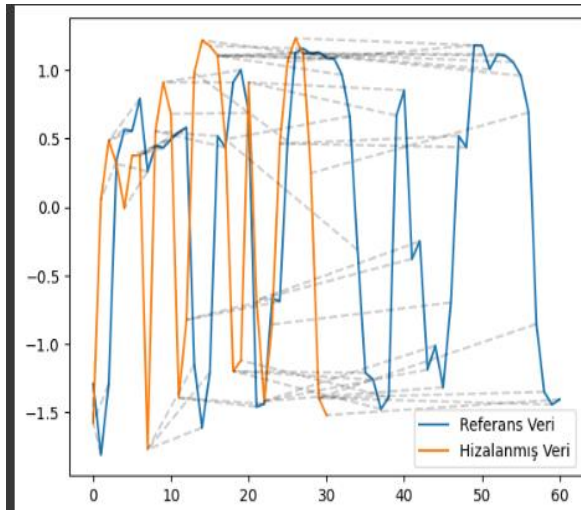
# İkinci veri kümesi
aligned_data = np.array([20.39, 112.95, 137.52, 127.82, 109.37, 131.21, 131.43, 9.43, 141.37, 161.67,
148.59, 30.77, 63.17, 165.63, 179.04, 176.81, 172.69, 138.01, 41.84, 46.31,
161.61, 71.27, 29.47, 61.22, 136.23, 170.18, 179.92, 173.6, 124.0, 30.76, 23.63])

# Veri kümesi normalize ediliyor
ref_data1 = (ref_data1 - np.mean(ref_data1)) / np.std(ref_data1)
aligned_data = (aligned_data - np.mean(aligned_data)) / np.std(aligned_data)

# DTW kullanarak hizalama yapılıyor
dist, cost, acc, path = dtw(ref_data1, aligned_data, dist=lambda x, y: np.linalg.norm(x - y))

# Hizalanmış veriler görselleştiriliyor
import matplotlib.pyplot as plt

plt.plot(ref_data1, label='Referans Veri')
plt.plot(aligned_data, label='Hizalanmış Veri')
for i, j in zip(path[0], path[1]):
    plt.plot([i, j], [ref_data1[i], aligned_data[j]], 'k--', alpha=0.2)
plt.legend()
plt.show()
```



2.Öğretmen ve öğrencileri video verisetinin karşılaştırılması

```
import numpy as np
!pip install dtw
from dtw import dtw
import matplotlib.pyplot as plt

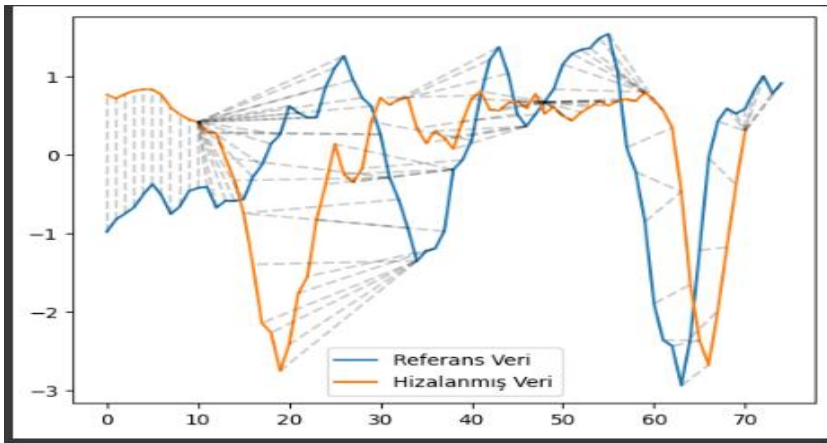
# Veri kümesi normalize ediliyor
ref_data = (ref_data - np.mean(ref_data)) / np.std(ref_data)
aligned_data7 = (aligned_data7 - np.mean(aligned_data7)) / np.std(aligned_data7)

# DTW kullanarak hizalama yapılıyor
dist, cost, acc, path = dtw(ref_data, aligned_data7, dist=lambda x, y: np.linalg.norm(x - y))

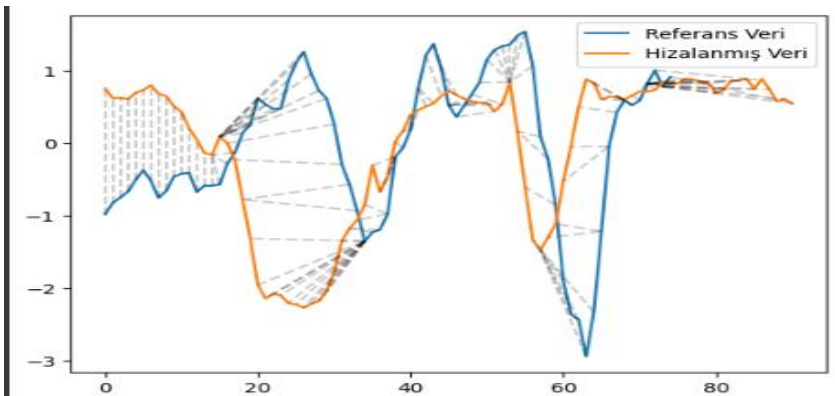
# Hizalanmış veriler görselleştiriliyor
import matplotlib.pyplot as plt

plt.plot(ref_data, label='Referans Veri')
plt.plot(aligned_data7, label='Hizalanmış Veri')
for i, j in zip(path[0], path[1]):
    plt.plot([i, j], [ref_data[i], aligned_data7[j]], 'k--', alpha=0.2)
plt.legend()
plt.show()
```

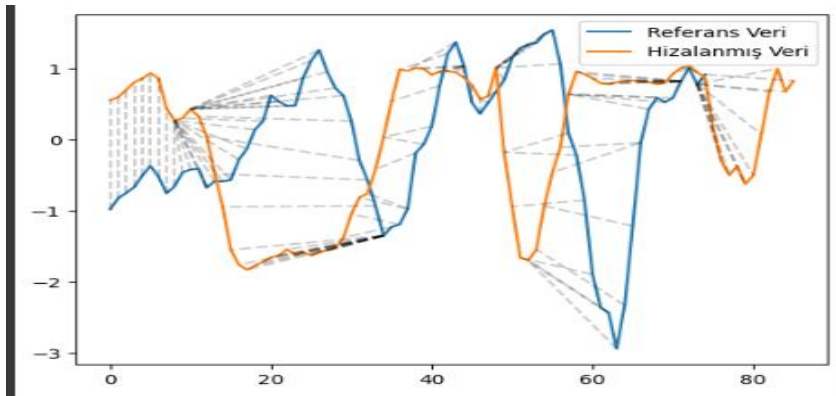
1.Align Data



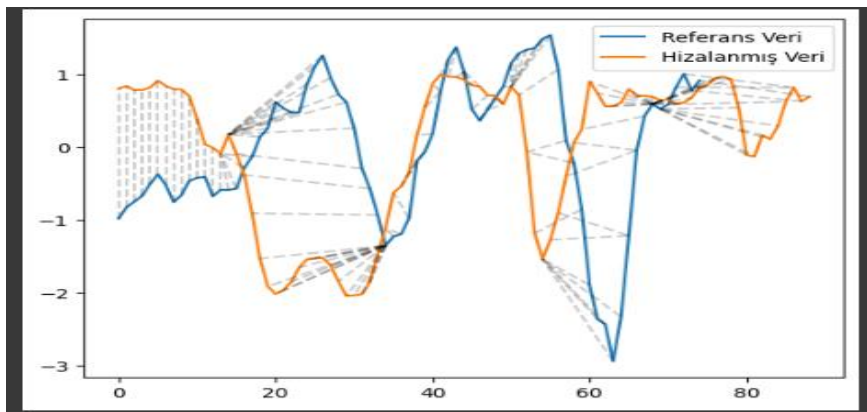
2.Align Data



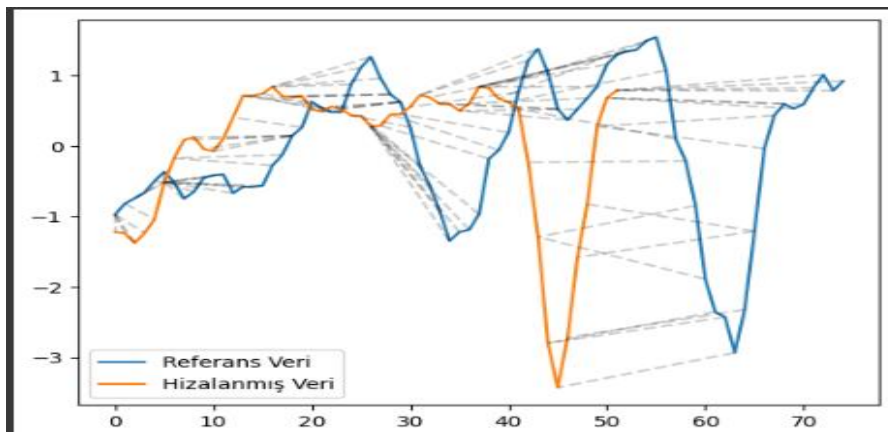
3.Align Data



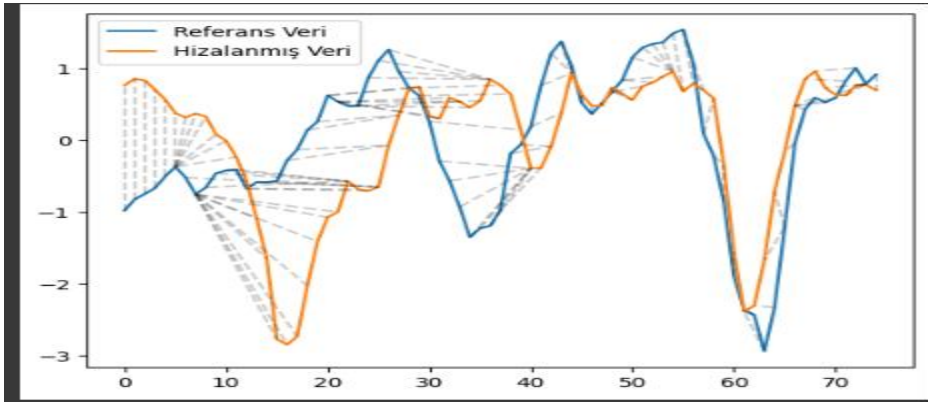
4.Align Data



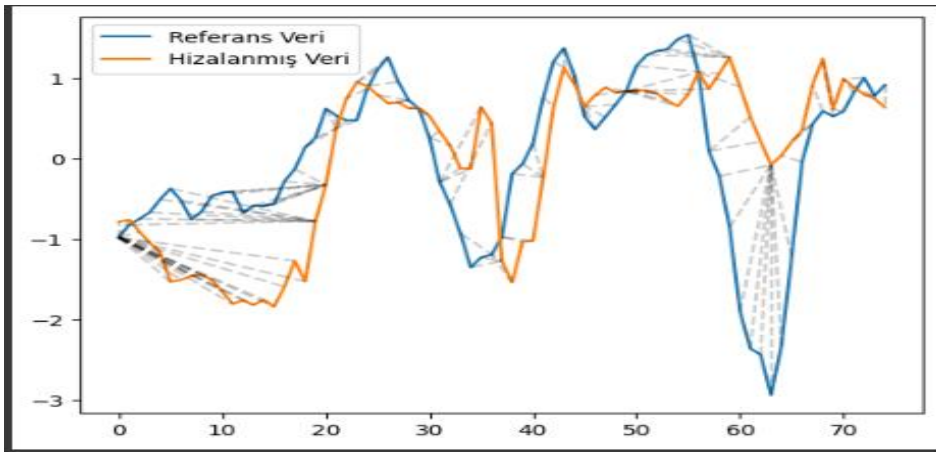
5.Align Data



6.Align Data



7.Align Data



Bu kod bloğu, bir referans veri kümesi ve bir hedef veri kümesi arasındaki zaman serilerinin hizalanmasını gerçekleştirir ve sonuçları görselleştirir. İlk olarak, her iki veri kümesi de normalize ediliyor. Normalizasyon, her bir veri kümesinin ortalama değerinin çıkarılması ve standart sapma ile bölünmesi işlemidir. Bu, her iki veri kümesinin de benzer bir ölekte olduğunu ve daha sonra gerçekleştirilecek hizalama işleminin daha doğru sonuçlar vermesini sağlar. Daha sonra, "DTW" (Dynamic Time Warping) adlı bir zaman serisi hizalama yöntemi kullanılarak hedef veri kümesi, referans veri kümesine hizalanır. DTW, iki zaman serisi arasındaki zaman ölçeği farklılıklarını ve sınırlarını dikkate alarak, iki zaman serisini hizalamak için bir dinamik programlama yöntemi kullanır. Bu, zaman serilerinin farklı hızlarda ilerlediği durumlarda bile doğru hizalamaların yapılmasını sağlar. Son olarak, hizalanmış veriler görselleştiriliyor. "Matplotlib" adlı bir grafik kütüphanesi kullanılarak, referans veri ve hizalanmış veri, ayrı ayrı çizdiriliyor. Ayrıca, hizalama işlemi sırasında oluşan yolu göstermek için, her iki zaman serisi arasındaki eşleştirmelerin bağlantılarını gösteren çizgiler çizdiriliyor. Bu kod bloğu, özellikle zaman serilerinin hizalanması veya benzerliklerinin ölçülmesi gerektiği durumlarda kullanılabilir. Örneğin, bir kişinin konuşmasının farklı bölümlerinin karşılaştırılması veya bir müzik parçasının farklı kayıtlarının karşılaştırılması gibi durumlarda kullanılabilir.