

# PROJE1\_4 RAPORU

## Giriş

Projenin amacı, OpenCV kullanarak birden çok videonun içindeki frame'leri çekip bilgisayarın o anki tarih-saat bilgisini yazmak ve bu karelerden yeni bir video oluşturmaktır.

## Gelişme

Projede, OpenCV kullanılarak giriş videoları kare kare okunmuş ve her kareye sistemden alınan anlık tarih-saat bilgisi eklenmiştir. Bu işlem 3 farklı video için tekrarlanarak her biri işlenmiş ve belirtilen klasöre çıktı olarak kaydedilmiştir. Kullanılan kod satırları aşağıdaki gibidir:

```
: # Gerekli kütüphaneler
import cv2
from datetime import datetime, timedelta
import os
```

Bu kod hücresinde, projenin çalışabilmesi için gerekli olan Python kütüphaneleri içe aktarılmıştır. cv2 modülü, OpenCV kütüphanesinin ana araçlarını kullanmamızı sağlar; bu sayede videoları okuyabilir, frame temelli işlemleri yapabilir ve görüntü üzerine metin ekleyebiliriz. datetime modülü, bilgisayarın o anki tarih ve saat bilgisine erişmek ve bunu belirli bir formatta yazdırmak için içe aktırılmıştır. os modülü ise videolarının kaydedileceği klasörün var olup olmadığını kontrol etmek veya yeni klasörler oluşturmak için eklenmiştir.

```
# Video dosyaları giriş-çıkış eşlemesi
video_files = [
    ("/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/inputv1.mp4", "/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/outputv1.mp4"),
    ("/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/inputv2.mp4", "/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/outputv2.mp4"),
    ("/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/inputv3.mp4", "/Users/kevserrkahraman/Desktop/VIDEO_PRPROJECT/videos/outputv3.mp4")
]

# Klasör yoksa oluşturmak için
os.makedirs("videos", exist_ok=True)
```

Bu kod bloğu, işlenecek giriş videoları ile elde edilecek çıkış videolarının yollarını tanımlar. video\_files listesi, her biri bir giriş videosu ve bu videodan üretilen çıkış videosunun dosya yolunu içeren ikililerden oluşur. Bu yapı sayesinde, her video için hangi dosyanın işleneceği ve çıktıların nereye kaydedileceği net bir şekilde belirlenmiş olur.

Ayrıca **os.makedirs("videos", exist\_ok=True)** satırı, videoların kaydedileceği videos klasörünün varlığını kontrol etmek için eklenmiştir. Eğer bu klasör mevcut değilse Python otomatik olarak oluşturur. exist\_ok=True parametresi ise klasör zaten mevcutsa hata vermesini engeller.

```
: # İşlenecek videoların giriş ve çıkış yollarını video_files listesinde tanımlama
for input_path, output_path in video_files:

    # Giriş videosunu açma
    cap = cv2.VideoCapture(input_path)

    # Video açılmadıysa uyarı ver ve diğer videoya geç
    if not cap.isOpened():
        print(f" Video açılmadı: {input_path}")
        continue
```

Bu kod parçası, video\_files listesinde tanımlanmış olan giriş ve çıkış video dosyaları üzerinde işlem yapmak için bir for döngüsü başlatır. Döngü her adımda, input\_path değişkenine giriş videosunun yolu, output\_path değişkenine ise işlenmiş videonun kaydedileceği yolu atar.

**cv2.VideoCapture(input\_path)** komutu, belirtilen giriş videosunu OpenCV aracılığıyla açar ve frame olarak okunabilir hale getirir. Bu adım, video üzerinde işlem yapabilmenin ilk aşamasıdır. Ardından **if not cap.isOpened():** kontrolü ile video dosyasının başarıyla açılıp açılmadığı test edilir. Eğer dosya açılmazsa ekrana hata mesajı yazdırılır.

```
# Video genişlik, yükseklik ve FPS değerlerini alma
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS)

# Çıkış videosu için codec (mp4v) tanımla
fourcc = cv2.VideoWriter_fourcc(*'mp4v')

# Çıkış videosunu oluşturma
out = cv2.VideoWriter(output_path, fourcc, fps, (width, height))

# Video karelerini tek tek işleme
while True:
    # Bir kare okuma
    ret, frame = cap.read()
    if not ret:
        # Kare yoksa video bitmiş, döngüden çıkma
        break
```

Bu kod bölümünde, video dosyasının teknik özelliklerini alıp çıktı videosunu oluşturmak ve frameleri tek tek işlemek için gerekli adımları içerir.

Öncelikle **cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH)** ve **cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)** komutlarıyla videonun piksel cinsinden genişlik ve yükseklik değerleri alınır. Bu değerler int() ile tam sayıya dönüştürülerek width ve height değişkenlerinde saklanır. **cap.get(cv2.CAP\_PROP\_FPS)** ile videonun saniyedeki frame sayısı elde edilir. Bu bilgiler, çıkış videosunun giriş videosu ile aynı çözünürlük ve FPS değerlerine sahip olması için gereklidir.

Ardından **cv2.VideoWriter\_fourcc(\*'mp4v')** komutu ile kullanılacak video **codec** belirlenir. Codec, ham video verilerini sıkıştırarak dosya boyutunu küçülten ve oynatma sırasında çözerek görüntü sağlayan bir algoritmadır. Burada seçilen **mp4v** codec'i, .mp4 uzantılı videolar için yaygın olarak kullanılan ve pek çok oynatıcıyla uyumlu bir formattır. Codec belirlendikten sonra **cv2.VideoWriter(output\_path, fourcc, fps, (width, height))** fonksiyonu ile çıkış videosu oluşturulur. Bu fonksiyon, çıkış dosyasının yolunu (output\_path), codec bilgisini (fourcc), saniyedeki frame sayısını ve çözünürlüğünü (width, height) parametreleri ile tanımlar. Böylece giriş videosuyla aynı özelliklere sahip ancak üzerine tarih-saat eklenmiş yeni bir video üretilmeye hazır hâle gelir.

Sonraki adımda **while True:** döngüsü ile giriş videosunun kareleri tek tek okunur. **cap.read()** komutu her çağrıldığında bir frame ve okuma işleminin başarılı olup olmadığını belirten **ret** değeri döner. Eğer **ret** False ise bu, videoda okunacak kare kalmadığı anlamına gelir ve **break** komutu ile döngü sonlandırılır. Bu yapı sayesinde video kareleri bitene kadar işlem yapılabilir.

```

# Video içinde geçen süre
elapsed_seconds = frame_index / fps
now = (start_time + timedelta(seconds=elapsed_seconds)).strftime("%Y-%m-%d %H:%M:%S")

# Tarih-saat bilgisini sol üst köşeye yazma
cv2.putText(frame, now, (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
            1, (0, 255, 0), 2)

# Kareyi çıkış videosuna yazma
out.write(frame)

# Kaynakları serbest bırakma
cap.release()
out.release()
print(f" Tamamlandı: {output_path}")

# Tüm videolar işlendikten sonra mesaj yazdırma
print(" Tüm videolar işlendi.")

```

Bu kod bölümünde, videonun o anki video içi zamanını hesaplayarak her kareye tarih-saat bilgisi ekleme ve işlenmiş kareleri yeni video dosyasına yazma işlemleri gerçekleştirilmektedir.

Öncelikle **elapsed\_seconds = frame\_index / fps** ifadesi ile, o an işlenen kare numarasının (**frame\_index**) video saniye hızına (fps) bölünmesi sonucu geçen süre saniye cinsinden hesaplanır. Bu süre, videonun başlangıç zamanı (**start\_time**) ile **timedelta(seconds=elapsed\_seconds)** aracılığıyla toplanarak videonun o kareye karşılık gelen kesin tarih-saat bilgisi elde edilir. Böylece yalnızca videonun süresi değil, bu sürenin gerçek zaman karşılığı da bulunmuş olur. Son olarak **strftime("%Y-%m-%d %H:%M:%S")** kullanılarak bu zaman bilgisi yıl-ay-gün saat:dakika:saniye formatına dönüştürülür ve kare üzerine yazılmaya hazır hâle getirilir.

Ardından **cv2.putText(...)** fonksiyonu kullanılarak bu bilgi, görüntünün sol üst köşesine yerleştirilir. Burada (10, 30) koordinatları metnin başlangıç konumunu, **cv2.FONT\_HERSHEY\_SIMPLEX** yazı tipini, 1 yazı boyutunu, (0, 255, 0) yeşil rengi ve 2 kalınlık değerini belirtir.

Metin eklenmiş kare **out.write(frame)** komutu ile çıkış videosuna kaydedilir. Döngü tamamlandığında **cap.release()** ve **out.release()** fonksiyonları kullanılarak video okuma ve yazma kaynakları serbest bırakılır.

```

Tamamlandı: /Users/kevserkahraman/Desktop/VIDEO_PROJECT/videos/outputv1.mp4
Tamamlandı: /Users/kevserkahraman/Desktop/VIDEO_PROJECT/videos/outputv2.mp4
Tamamlandı: /Users/kevserkahraman/Desktop/VIDEO_PROJECT/videos/outputv3.mp4
Tüm videolar işlendi.

```

Bu çıktılar, her bir giriş videosunun başarıyla işlenip üzerine tarih-saat bilgisi eklendiğini ve belirtilen dosya yoluna kaydedildiğini gösterir.

## SONUÇ

Bu çalışmada, OpenCV kütüphanesi kullanılarak üç farklı giriş videosunun her karesine anlık tarih-saat bilgisi eklenmiş ve bu kareler yeni birer .mp4 dosyası olarak kaydedilmiştir.. Tüm giriş videoları sorunsuz şekilde işlenmiş, belirtilen yollar altında yeni çıkış videoları oluşturulmuş ve her video için ayrı ayrı başarı mesajı, ayrıca tüm işlemlerin tamamlandığını belirten genel bir mesaj ekrana yazdırılmıştır.

## KAYNAKÇA

[https://docs.opencv.org/4.x/dd/d43/tutorial\\_py\\_video\\_display.html](https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html)