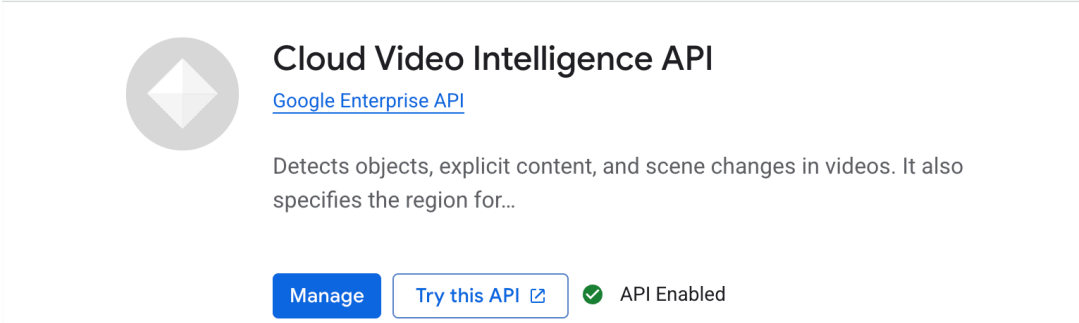
Proje 6 - Video Akışı ve İşleme Uygulaması



Genel Bilgi

Bu projede, canlı bir video yayını üzerinde gerçek zamanlı analiz yapabilen bir sistem geliştirilmiştir. Kullanıcılar, yayınlarını RTMP protokolü üzerinden OBS (Open Broadcaster Software) ile başlatabilmekte ve yayın esnasında sunucu tarafından alınan video parçaları, Google Cloud Video Intelligence API kullanılarak analiz edilmektedir. Analiz sonucunda, videolardaki içerikler etiketlenmekte ve anlamlı bilgiler çıkarılmaktadır. Sistem Node.js altyapısıyla geliştirilmiş olup, FFmpeg ile video parçalama, Express.js ile API yönetimi ve Google Cloud ile video analizi sağlanmıştır. Projenin amacı, gerçek zamanlı video verilerinin otomatik olarak işlenmesi ve anlamlandırılması sürecini kolaylaştıran bir prototip sunmaktır.

Kodların Açıklamaları

İndex.js

Sunucu ve Google Cloud İstemcisi Tanımı:

```
const app = express();
const PORT = 3000;
const client = new VideoIntelligenceServiceClient({ keyFilename: 'gcp-key.json' });
```

`app`: Express uygulamasını başlatır.

`PORT`: Sunucunun dinleyeceği port numarasıdır.

client: GCP Video Intelligence API'ye bağlanmak için oluşturulan istemcidir. Kimlik doğrulaması için gcp-key.json dosyası kullanılır.

Multer ile Dosya Yükleme Ayarı:

```
const upload = multer({ dest: 'uploads/' });
```

Bu ayar, kullanıcıdan alınan video dosyalarının geçici olarak uploads/ klasörüne kaydedilmesini sağlar.

Video Etiketleme (Label Detection) İşlemi:

```
const [operation] = await client.annotateVideo({
  inputContent,
  features: ['LABEL_DETECTION'],
});

const [response] = await operation.promise();
const labels = response.annotationResults[0].segmentLabelAnnotations;
```

API çağrısı yapılır ve LABEL_DETECTION özelliği istenir.
İşlem tamamlandıktan sonra dönen etiket verileri labels değişkeninde toplanır.

Etiketlerin Formatlanması:

```
const result = labels.map(label => ({
  description: label.entity.description,
  startTime: label.segments[0].segment.startTimeOffset.seconds || 0,
  endTime: label.segments[0].segment.endTimeOffset.seconds || 0,
}));
```

API'nin döndürdüğü karmaşık veri, sade bir formatta düzenlenerek description, startTime, endTime bilgileri çıkarılır.

watchers.js

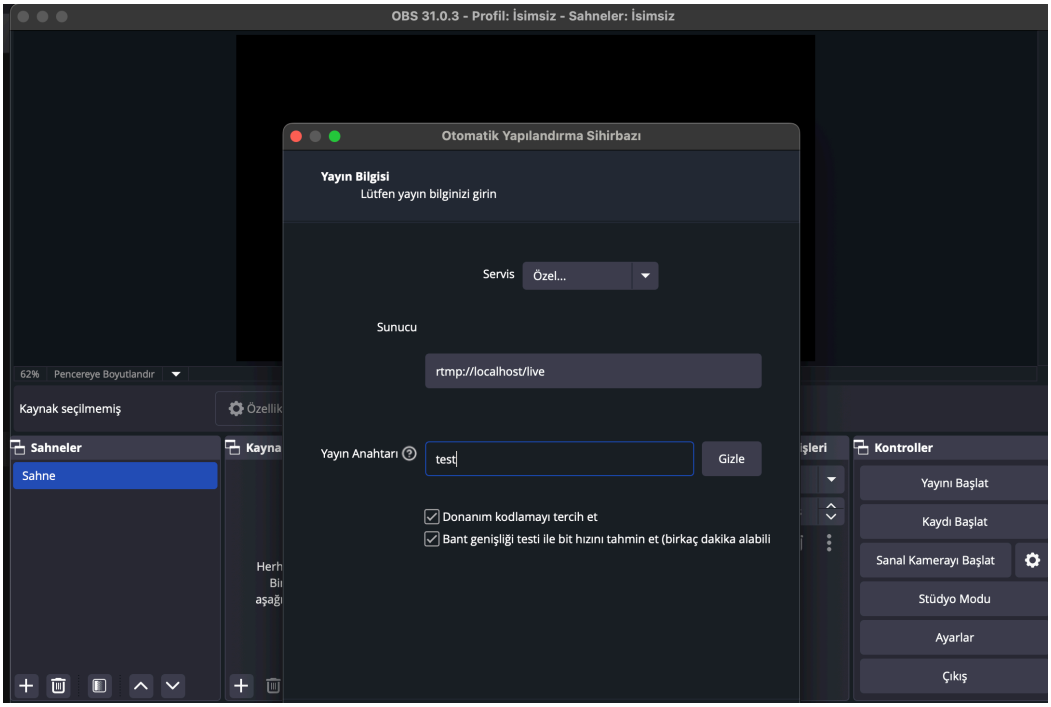
`watchers.js` dosyası, canlı yayından elde edilen video parçalarını otomatik olarak analiz etmek amacıyla kullanılan bir betiktir. Belirli bir klasöre (örneğin `segments/`) kaydedilen her yeni `.mp4` dosyasını algılayarak, bu dosyaları `index.js` içerisinde tanımlanan `/api/analyze` endpoint'ine HTTP POST isteği ile gönderir. Bu sayede, kullanıcı müdahalesine gerek kalmadan her video parçası sırasıyla analiz edilir. API'den gelen analiz sonuçları kaydedilebilir ya da konsola yazdırılabilirken, işlenen dosyalar sistem yükünü azaltmak amacıyla genellikle silinir. Bu otomasyon, canlı video analiz süreçlerinin kesintisiz ve verimli şekilde işlemesini sağlar.

rtmp-server.js

`rtmp-server.js` dosyası, canlı video yayını alabilmek için Node.js tabanlı bir RTMP (Real-Time Messaging Protocol) sunucusu kurar. Bu dosyada `Node-Media-Server` kütüphanesi kullanılarak RTMP protokolünü destekleyen bir yayın altyapısı oluşturulur. Yayıncı (örneğin OBS gibi bir uygulama), bu sunucuya RTMP üzerinden canlı yayın gönderdiğinde, `rtmp-server.js` bu yayını alır ve yayını bir endpoint altında kullanılabilir hale getirir (örneğin: `rtmp://localhost/live/test`). Bu yapı sayesinde canlı yayın akışı sistem tarafından yakalanarak daha sonra analiz edilmek üzere kaydedilebilir veya farklı işlemlere yönlendirilebilir. RTMP sunucusu bu projenin canlı video analiz sürecinin temelini oluşturur.

OBS Studio Ayarları

Projede canlı yayını



başlatmak ve RTMP sunucusuna video akışı göndermek için OBS Studio kullanılmıştır. OBS üzerinde yayına başlamadan önce, "Ayarlar > Yayın" menüsüne gidilerek Yayın Türü "Özel..." olarak seçilmiş, Sunucu kısmına `rtmp://localhost/live` ve Akış Anahtarı olarak `test` yazılmıştır. Bu ayarlar sayesinde OBS, yerel olarak çalışan RTMP sunucusuna `rtmp://localhost/live/test` adresi üzerinden video yayını gönderebilmiştir. Yayına başlamadan önce, OBS arayüzündeki "Kaynaklar" bölümüne bir ekran görüntüsü, video veya medya kaynağı eklenmesi gerekmektedir; aksi halde boş bir ekran yayına aktarılır. Bu yapı, canlı video akışının düzgün bir şekilde backend tarafından alınması ve analiz edilmesi için temel bir adımdır.

Proje Nasıl Çalışır

Sırası ile backend klasöründe aşağıdaki adımlar uygulanır

node index.js

node rtmp-server.js

Bu adımdan sonra OBS Studio üzerinden canlı yayın açılır

ffmpeg -i rtmp://localhost/live/test -c copy -f segment -segment_time 30 -reset_timestamps 1 uploads/part_%03d.mp4

node watchers.js

Yukarıdaki adınları uyguladıktan sonra video analiz sonuçları terminale json olarak yazdırılır

```
part_003.mp4 için analiz sonucu: [
  {
    description: 'shark',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 35, high: 0, unsigned: false }
  },
  {
    description: 'wildlife',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 35, high: 0, unsigned: false }
  },
  {
    description: 'animal',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 35, high: 0, unsigned: false }
  }
]
part_004.mp4 için analiz sonucu: [
  {
    description: 'wilderness',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 25, high: 0, unsigned: false }
  },
  {
    description: 'wildlife',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 25, high: 0, unsigned: false }
  },
  {
    description: 'giraffe',
    startTime: { low: 0, high: 0, unsigned: false },
    endTime: { low: 25, high: 0, unsigned: false }
  },
  {
    description: 'terrestrial animal',
    startTime: { low: 0, high: 0, unsigned: false },
```