

SEAL Video Automation System 🚀

📋 Sistem Özeti

Bu sistem, Self-Evolving AI Layer (SEAL) mimarisiyle çalışan, tamamen otomatik video içerik üretimi ve yayınlama platformudur. Sistem:

- **Trend analizi** ile popüler konular bulur
- **AI ile senaryo** üretir (Ollama + Mistral)
- **Çoklu dilde seslendirme** yapar (ElevenLabs)
- **Otomatik video** oluşturur
- **YouTube Shorts ve TikTok'a** yükler
- **Affiliate linkler** ekler
- **Kullanıcı geri bildirimiyle** kendini geliştirir
- **Eğitim paketleri** oluşturur

🏗️ Sistem Mimarisi

Trend Analizi → AI Senaryo → TTS → Video → Upload → Feedback → SEAL Learning
↓ ↓ ↓ ↓ ↓ ↓ ↓
Google Trends → Mistral → ElevenLabs → FFmpeg → APIs → Analytics → Fine-tune

📁 Proje Yapısı

seal-video-automation/

```
|─── 📁 core/
|   |─── generate_script.py    # LangChain + RAG ile metin üretimi
|   |─── trend_analyzer.py    # Google Trends + Reddit API
|   |─── voiceover.py         # ElevenLabs çoklu dil TTS
|   |─── create_video.py      # FFmpeg video üretimi + altyazı
|   |─── upload.py           # YouTube + TikTok yükleme
|   |─── affiliate_manager.py # Otomatik affiliate link ekleme
|   |─── feedback_collector.py # Metrik toplama ve analiz
|   └─── course_builder.py   # Eğitim paketi oluşturma
|─── 📁 data/
|   |─── knowledge_base/      # RAG için veri
|   |─── feedback_logs/       # Kullanıcı geri bildirim verileri
|   └─── trends/              # Trend analizi sonuçları
|─── 📁 fine_tune/
|   |─── axolotl_config.yml   # Mini fine-tune ayarları
|   └─── lora_adapter/        # LoRA modeli
|─── 📁 templates/
|   |─── video_backgrounds/   # Stok video şablonları
|   |─── audio_templates/     # Ses efektleri
|   └─── subtitle_styles/     # Altyazı stil dosyaları
|─── 📁 automation/
|   |─── n8n_workflows/       # n8n iş akışları
|   └─── make_scenarios/      # Make.com senaryoları
|─── 📁 gcloud/
|   |─── deploy.sh           # Google Cloud deploy scripti
|   |─── dockerfile          # Container yapılandırması
|   └─── cloud-functions/    # Serverless fonksiyonlar
|─── docker-compose.yml
|─── requirements.txt
|─── .env.example
|─── README.md
```

🔧 Gereksinimler

Yazılım Gereksinimleri

- Python 3.9+
- Docker & Docker Compose
- FFmpeg
- Ollama

API Anahtarları

- ElevenLabs API Key

- YouTube Data API v3
- Google Trends API (opsiyonel)
- Reddit API (opsiyonel)
- TikTok API (3rd party)

Kurulum (Local)

1. Depoyu Klonlayın

```
bash

git clone https://github.com/your-username/seal-video-automation.git
cd seal-video-automation
```

2. Sanal Ortam Oluşturun

```
bash

python -m venv venv
source venv/bin/activate # Linux/Mac
# venv\Scripts\activate # Windows
```

3. Bağımlılıkları Yükleyin

```
bash

pip install -r requirements.txt
```

4. Ollama Kurulumu

```
bash

# Ollama'yı indirin ve kurun
curl -fsSL https://ollama.ai/install.sh | sh

# Mistral modelini çekin
ollama pull mistral
```

5. Ortam Değişkenlerini Ayarlayın

```
bash

cp .env.example .env
# .env dosyasını editleyip API anahtarlarınızı girin
```

`.env` örneği:

```
env

# AI Model Ayarları
OLLAMA_BASE_URL=http://localhost:11434
OLLAMA_MODEL=mistral

# TTS Ayarları
ELEVENLABS_API_KEY=your_elevenlabs_key_here
ELEVENLABS_VOICE_ID=21m00Tcm4TlvDq8ikWAM

# YouTube API
YOUTUBE_API_KEY=your_youtube_api_key
YOUTUBE_CHANNEL_ID=your_channel_id

# Trend Analysis
GOOGLE_TRENDS_API_KEY=your_trends_api_key
REDDIT_CLIENT_ID=your_reddit_client_id
REDDIT_CLIENT_SECRET=your_reddit_secret

# Affiliate Settings
AMAZON_AFFILIATE_TAG=your_amazon_tag
AFFILIATE_LINKS_FILE=data/affiliate_links.json

# RAG Ayarları
CHROMA_PERSIST_DIRECTORY=./data/vector_db
```

6. İlk Çalıştırma

```
bash

# Sistem kontrolü
python core/system_check.py

# Test video üretimi
python core/main.py --test-mode
```

Google Cloud Deploy

1. Google Cloud Projesini Hazırlayın

```
bash
```

Google Cloud CLI kurulumu

`curl https://sdk.cloud.google.com | bash`

`exec -l $SHELL`

`gcloud init`

Proje oluşturma

`gcloud projects create seal-video-automation --name="SEAL Video Automation"`

`gcloud config set project seal-video-automation`

Gerekli API'leri etkinleştirin

`gcloud services enable compute.googleapis.com`

`gcloud services enable cloudfunctions.googleapis.com`

`gcloud services enable storage.googleapis.com`

2. Compute Engine VM Oluşturma

`bash`

VM oluşturma (4 vCPU, 16GB RAM)

`gcloud compute instances create seal-automation-vm \`

`--image-family=ubuntu-2004-lts \`

`--image-project=ubuntu-os-cloud \`

`--machine-type=n1-standard-4 \`

`--boot-disk-size=50GB \`

`--zone=us-central1-a \`

`--tags=http-server,https-server`

VM'e bağlanma

`gcloud compute ssh seal-automation-vm --zone=us-central1-a`

3. VM'de Kurulum

`bash`

VM içinde

`sudo apt update && sudo apt upgrade -y`

`sudo apt install docker.io docker-compose python3-pip git -y`

Projeyi klonlayın

`git clone https://github.com/your-username/seal-video-automation.git`

`cd seal-video-automation`

Docker ile çalıştırın

`sudo docker-compose up -d`

4. Cloud Functions ile Otomasyon

bash

Trend analizi fonksiyonu deploy

```
gcloud functions deploy trend-analyzer \
  --source=gcloud/cloud-functions/trend-analyzer \
  --entry-point=main \
  --runtime=python39 \
  --trigger-http \
  --allow-unauthenticated
```

Günlük içerik üretimi için cron job

```
gcloud scheduler jobs create http daily-content-generation \
  --schedule="0 8 * * *" \
  --uri="https://us-central1-seal-video-automation.cloudfunctions.net/trend-analyzer" \
  --http-method=POST
```

Kullanım

Manuel Video Üretimi

bash

Tek video üretimi

```
python core/main.py --topic "How to make money with AI in 2025"
```

Çoklu dil üretimi

```
python core/main.py --topic "Excel automation tips" --languages "en,de,ko,zh"
```

Affiliate linklerle

```
python core/main.py --topic "Best AI tools 2025" --include-affiliates
```

Otomatik İçerik Üretimi

bash

Trend tabanlı otomatik üretim

```
python core/auto_generator.py --count 5 --schedule daily
```

Otomasyon başlatma (24/7 çalışır)

```
python core/automation_daemon.py
```

Eğitim Paketi Oluşturma

bash

Mevcut videolardan kurs oluşturma

```
python core/course_builder.py --topic "YouTube AI Automation" --video-count 20
```

SEAL Öğrenme Sistemi

Geri Bildirim Toplama

Sistem otomatik olarak şu metrikleri toplar:

- YouTube: İzlenme, beğeni, yorum sayısı
- TikTok: Görüntülenme, beğeni, paylaşım
- Kullanıcı geri bildirimleri

Mini Fine-Tune

bash

Aylık model güncelleme

```
python fine_tune/mini_finetune.py --feedback-threshold 100
```

Manuel fine-tune

```
python fine_tune/manual_finetune.py --data-path data/feedback_logs/positive_examples.json
```

İzleme ve Analiz

Metrik Dashboard

bash

Gradio dashboard başlatma

```
python core/dashboard.py
```

<http://localhost:7860> adresinden erişim

Log Analizi

bash

Performans raporları

```
python core/analytics.py --report-type weekly
```

```
python core/analytics.py --report-type affiliate-performance
```

Özelleştirme

Yeni Dil Ekleme

1. `core/voiceover.py`'da yeni dil kodunu ekleyin

2. ElevenLabs'de uygun sesi seçin
3. Altyazı şablonunu `templates/subtitle_styles/` klasörüne ekleyin

Yeni Video Şablonu

1. `templates/video_backgrounds/` klasörüne arka plan ekleyin
2. `core/create_video.py`'da şablon ayarlarını güncelleyin

Affiliate Link Yönetimi

`data/affiliate_links.json` dosyasını düzenleyin:

```
json
{
  "ai_tools": {
    "ChatGPT Plus": "https://chatgpt.com/plus?ref=your_code",
    "Midjourney": "https://midjourney.com?ref=your_code"
  },
  "education": {
    "Coursera": "https://coursera.org?ref=your_code"
  }
}
```

💰 Monetizasyon Özellikleri

Otomatik Affiliate Ekleme

- Video içeriğine uygun ürün önerileri
- Açıklama kısmına otomatik link ekleme
- Performans takibi

Kurs Oluşturma

- Videolardan otomatik PDF oluşturma
- Gumroad/Teachable entegrasyonu
- E-posta listesi toplama

Çoklu Platform

- YouTube Shorts (AdSense geliri)
- TikTok Creator Fund
- Instagram Reels

🔧 Sorun Giderme

Yaygın Hatalar

bash

Ollama bağlantı hatası

`sudo systemctl start ollama`

ElevenLabs API limiti

.env dosyasında alternatif TTS ayarlayın

FFmpeg hatası

`sudo apt install ffmpeg`

Docker izin hatası

`sudo usermod -aG docker $USER`

Log Kontrolü

bash

Sistem logları

`tail -f logs/system.log`

Hata logları

`tail -f logs/error.log`

Upload logları

`tail -f logs/upload.log`



Performans Optimizasyonu

GPU Hızlandırması

bash

NVIDIA GPU desteği

`sudo apt install nvidia-docker2`

docker-compose-gpu.yml kullanın

Paralel İşleme

python

config.py içinde

`PARALLEL_VIDEOS = 3`

`CONCURRENT_UPLOADS = 2`

Güvenlik

API Anahtarları

- `.env` dosyasını asla commit etmeyin
- Google Cloud Secret Manager kullanın
- API anahtarlarını düzenli olarak rotate edin

Veri Koruması

- Kullanıcı verilerini şifreleyin
- GDPR uyumluluğunu sağlayın
- Backup stratejisi uygulayın

Kaynaklar ve Referanslar

- [Ollama Documentation](#)
- [ElevenLabs API](#)
- [YouTube Data API](#)
- [LangChain Documentation](#)
- [Google Cloud Documentation](#)

Katkıda Bulunma

1. Fork yapın
2. Feature branch oluşturun (`git checkout -b feature/amazing-feature`)
3. Commit yapın (`git commit -m 'Add amazing feature'`)
4. Push yapın (`git push origin feature/amazing-feature`)
5. Pull Request oluşturun

Lisans

Bu proje MIT lisansı altında lisanslanmıştır. Detaylar için `LICENSE` dosyasına bakın.

Destek

Herhangi bir sorun yaşarsanız:

- GitHub Issues kullanın
 - Dokümantasyonu kontrol edin
 - Community Discord'a katılın
-

Not: Bu sistem sürekli geliştirilmektedir. Güncellemeler için repository'yi star'layın ve watch edin!