



mirzabuilds
@MIRZABUILDS



ÜCRETSİZ REHBER · AÇIK AĞIRLIK

Video modelini kendi makinende çalıştır

LTX-2, videoyu ve senkron sesi tek geçište üreten açık ağırlıklı bir model. Ağırlıkları indiriyorsun, kendi bilgisayarında çalışıyor. Kredi yok, kuyruk yok, kare başına ücret yok.

Bu rehber kurulumdan ilk klibe, oradan kodlama ajanına sahne kurgulatmaya kadar tüm zinciri veriyor. Komutların tamamı Lightricks'in kendi deposundan doğrulandı. Donanım beklentisini ve lisans şartını da saklamadan yazdım, çünkü bu modelin asıl maliyeti para değil, GPU.

Tek model

Video ve ses aynı geçište, aynı birleştirme adımı yok

Sınırsız deneme

Bulut kredisi yaktmadığın için aynı sahneyi on kez üretebilirsin

ÖZET

Beş Adım

Sırayla git. İkinci adım en uzun süreni, **onlarca GB** ağırlık indiriyorsun. Beşinci adım ise işi tek klipten gerçek bir videoya çeviren yer.

#	ADIM
01	Depoyu kur
02	Şartları kabul et, ağırlıkları indir
03	İlk klibi üret
04	Belleği kendi makinene göre ayarla
05	Sahneleri kodlama ajanına kurgulat

MODEL

Tek Model, Altı Ayrı İş

LTX-2, videoyu ve sesi ayrı üretilip birleştiren bir zincir değil. İkisini aynı geçište üreten tek bir model. Depodaki hazır pipeline'lar şunları veriyor.

Yazıdan video

Prompt yazarsın, senkron sesli klip çıkar. En hızlı yol bu.

```
ltx_pipelines.distilled
```

Görselden video

Elindeki kareyi başlangıç yapar, hareketi model kurar.

```
--image bayırağı
```

Üretim kalitesi

Aynı damıtlılmış model, üstüne detaylandırma katmanı. Yavaş ama temiz.

```
ltx_pipelines.dfr_pipeline
```

Sesten video

Var olan bir ses dosyasına görüntü ürettirirsin.

```
ltx_pipelines.a2vid_two_stage
```

Kare arası doldurma

İki anahtar kare verirsın, aradaki hareketi model tamamlar.

```
ltx_pipelines.keyframe_interpolation
```

Yeniden çekim

Bitmiş videonun sadece bir zaman aralığını yeniden üretir.

```
ltx_pipelines.retake
```

İPUÇU

Hepsi aynı ağırlıklıları kullanıyor. Bir kez indirdikten sonra pipeline değiştirmek sadece komut değiştirmek demek.

KURULUM

Adım Adım

Her adımda dört şey var: **ne yaptığı, çalıştıracağın komut, ipucu ve o adının tuzağı**. Komutlar github.com/Lightricks/LTX-2 deposundan alındı ve LTX-2.3 sürümüne göre yazıldı.

ADIM 01 Depoyu kur

Ağırlıkları indirmeden önce kurulumun sorunsuz bittiğini gör. Bu adım birkaç dakika, sonraki adım onlarca gigabayt.

TERMINAL

```
git clone https://github.com/Lightricks/LTX-2.git
cd LTX-2
uv sync --extra natten
```

NE YAPIYORSUN

- Depoyu klonla
- Klasöre gir
- Bağımlılıkları kur

İPUÇU

natten yalnızca Linux ve CUDA'da kuruluyor. Windows ve macOS'ta otomatik atlanıyor, çözümleme daha yavaş bir yola düşüyor ama komut aynı kalyor.

TUZAK

uv kurulu değilse komut hata verir. Önce uv'yi kur, sonra bu adıma dön.

ADIM 02 Şartları kabul et, ağırlıkları indir

Ağırlıklı Hugging Face'te ve kapılı. İndirmeden önce model sayfasındaki şartları kabul edip giriş yapman gerekiyor. İki komut, onlarca gigabayt.

TERMINAL

```
hf auth login

# LTX-2.3 tek dosya: transformer, VAE'ler ve metin projeksiyonu bir arada
hf download Lightricks/LTX-2.3 \
  ltx-2.3-22b-distilled-1.1.safetensors \
  ltx-2.3-spatial-upscaler-x2-1.1.safetensors \
  --local-dir models/ltx-2.3

# metin kodlayıcı ayrı iniyor, 2.3 için Gemma 3
hf download google/gemma-3-12b-it-qat-q4_0-unquantized \
  --local-dir models/gemma-3-12b
```

NE YAPIYORSUN

- Hugging Face'te LTX-2.3 sayfasındaki şartları onayla
- Read yetkili bir token ile giriş yap
- Damıtlılmış checkpoint ve upscaler'ı models/ltx-2.3 altına indir
- Gemma 3'ü ayrı bir klasöre indir

İPUÇU

2.3'te transformer, video ve ses VAE'leri, metin projeksiyonu tek dosyanın içinde geliyor. Ayrı ayrı indirmen gereken tek şey metin kodlayıcı.

TUZAK

401 ya da 403 alırsan token'ın bozuk demek değil, model kapılı. Şartları kabul et. İnce ayarlı token kullanıyorsan "read gated repos" iznini ayrıca açman gerekiyor.

ADIM 03 İlk klibi üret

Damıtlılmış pipeline en hızlısı: sabit sekiz adımlık program kullanıyor ve ayrıca yönlendirme istemiyor. Başlangıç için doğru yer burası.

TERMINAL

```
uv run python -m ltx_pipelines.distilled \
  --distilled-checkpoint-path models/ltx-2.3/ltx-2.3-22b-distilled-1.1.safetensors \
  --spatial-upscaler-path models/ltx-2.3/ltx-2.3-spatial-upscaler-x2-1.1.safetensors \
  --gemma-root models/gemma-3-12b \
  --num-frames 121 \
  --seed 42 \
  --output-path çıktı.mp4 \
  --prompt "Yağmurdan sonra ıslak asfaltta yürüyen orta yaşlı bir adam, omuz hizasından takip eden kamera, neon tabelaların yansımaları zeminde titriyor, sig alan derinliği, soğuk mavi tonlar, sabit ritim"
```

NE YAPIYORSUN

- Kare sayısını --num-frames belirler
- Çözünürlüğü --height ve --width belirler, ikisi de 64'e bölünebilmeli
- seed aynı kaldığı sürece aynı prompt aynı sonucu verir

İPUÇU

İlk denemede güzel video peşinde koş. Düşük çözünürlük, az kare, sabit seed. Amaç zincirin çalıştığını kanıtlamak.

TUZAK

--height ve --width 64'e bölünebilmek zorunda. 4K çıktı 3840×2160 değil, 3840×2176. 2160 o ızgaraya oturmuyor ve komut hata veriyor.

ADIM 04 Belleği kendi makinene göre ayarla

Belleğin yetmediğinde model çalışmaz değil, yavaşlar. İki bayrak var ve ikisi de her pipeline'da çalışıyor.

TERMINAL

```
uv run python -m ltx_pipelines.distilled \
  --distilled-checkpoint-path models/ltx-2.3/ltx-2.3-22b-distilled-1.1.safetensors \
  --spatial-upscaler-path models/ltx-2.3/ltx-2.3-spatial-upscaler-x2-1.1.safetensors \
  --gemma-root models/gemma-3-12b \
  --image "kaneler/urun.png" \
  --num-frames 121 --seed 42 --output-path urun.mp4 \
  --quantization fp8-cast --offload cpu \
  --prompt "Ürün yavaşça kendi ekseninde dönüyor, yumuşak stüdyo ışığı"
```

```
# hangi bayraklar var, tam liste:
uv run python -m ltx_pipelines.distilled --help
```

NE YAPIYORSUN

- quantization fp8-cast modeli FP8'e düşürüyor
- offload cpu ağırlıkları sistem belleğine taşıyor
- offload disk RAM de yetmediğinde diskten akıtıyor, en yavaşı
- image ile elindeki kareyi başlangıç yaparsın

İPUÇU

Kendi makinenin sınırını bir kez bul ve not et. Her seferinde yeniden denemek yerine çalışan kombinasyonu kaydet.

TUZAK

Bayrakları hafızandan yazma. Sürümler arası isimler değişiyor, --help çıktısı her zaman doğruyu söylüyor.

ADIM 05 Sahneleri kodlama ajanına kurgulat

Model tek seferde kısa klip üretiyor. Uzun videoyu sahne sahne üretilip birleştirerek elde ediyorsun, ve bu zinciri elle sürmek yerine ajana verirsın.

NE YAPIYORSUN

- Ajana brief'i ver, önce plan.md yazsın
- Planı onayla, üretime ondan sonra geçsin
- Her sahne için ayrı bir CLI çağırısı yapısın
- Devamlılık için bir sahnenin son karesini sonraki çağrıya --image olarak versin
- Bitince ffmpeg ile birleştirsin

İPUÇU

LTX-2 bir komut satırı aracı. Dosya okuyup komut çalıştırabilen her ajan onu sürebilir: Claude Code, Codex, Cursor fark etmiyor.

TUZAK

Ajanı doğrudan üretime salarsan yirmi dakikayı yanlış sahnelere harcarsın. Plan dosyası ucuz, üretim pahalı. Önce metni onayla, sonra kareyi üret.

ŞABLON

Ajana Vereceğin Brief

Beşinci adımın karşılığı bu. Köşeli parantezleri kendi işine göre doldur, gerisi olduğu gibi kalsın. Ajan önce plan yazacak, sen onaylayınca üretecek.

```
# Rol: LTX-2 ile sahne sahne video üreten yapısın

Depo kökündesin (Lightricks/LTX-2). Ağırlıklar models/ altında.
```

Görev

Konu: [konuyu buraya yaz]
Toplam süre: [örn. 30 saniye]
Ton: [örn. sinematik, sakin]
En boy oranı: [16:9 veya 9:16]

Nasıl çalış

- Önce plan.md yaz. Her sahne için süre, kamera, ışık ve tek paragraflık görsel prompt olsun. Onayımı al, üretime geçme.
- Onaydan sonra her sahne için ltx_pipelines.distilled çağır.
- Çıktıları: sahneler/01.mp4, 02.mp4 diye kaydet.
- Devamlılık için bir sahnenin son karesini çıkar ve sonraki çağrıya --image olarak ver.
- Kullandığın --seed değerlerini plan.md'ye yaz. Beğenmediğin sahneyi aynı seed ile tek başına yeniden üretebilmeyim.
- Bitince ffmpeg ile birleştir, tek dosya ver.

Kurallar

- Bayrak uydurma. Emin değilsen --help çıktısını oku.
- Tüm sahneleri üretmeden önce tek sahnede dene, onaylayınca devam et.
- Bellek hatasından --quantization fp8-cast --offload cpu ekle ve çözünürlüğü düşür. Sessizce sahne atlama.
- Her sahnenin süresini ve seed'ini rapor et.

İPUÇU

Seed'leri plana yazdırman önemli. Beğenmediğin tek sahneyi baştan üretirsin, videonun tamamını değil.

PROMPT

Görüntü Yönetmeni Gibi Yaz

Lightricks'in kendi rehberi net: tek akan paragraf, kronolojik sıra, 200 kelimeyi geçme. Madde madde sıfat yığmak değil, çekimli tarif etmek işe yarıyor.

01 Ana eyleme başla

İlk cümle ne olduğunu söylesin. Sahneyi kurmadan önce hareketi ver.

02 Hareketi ve görünüşü netleştir

Jestler, yön, hız. Karakterin ve nesnelerin görünümünü kesin tarif et.

03 Arka planı, kamerayı ve ışığı yaz

Ortam, kamera açısı ve hareketi, ışık ve renkler. Üçü de açıkça geçsin.

04 Değişimi sona koy

Klip içinde bir şey değişiyorsa, ani bir olay ya da ışık dönüşü, onu sona ekle.

05 Sesi de aynı paragrafta tarif et

Model sesi ayrı üretmiyor. Replik istiyorsan cümleyi tırnak içinde ver ve sesin tonunu yaz.

İPUÇU

Prompt yazmakta zorlanırsan --enhance-prompt bayrağı modelin kendi zenginleştiricisini devreye sokuyor.

SÜRÜM

2.5'e geçecek olursan

Depo artık LTX-2.5'i öneriyor ve 2.3'ü "legacy" başlığı altında tutuyor. İki sürüm arasındaki fark dosya düzeni: 2.3 tek dosya, transformer ve VAE'ler onun içinde, metin kodlayıcı ayrı iniyor. 2.5 ise bileşen başına ayrı dosya veriyor ve Gemma 4'ü ağırlıkların içinde getiriyor, ayrı indirme yok. Bu yüzden bayraklar da ayrışıyor: 2.3'te --distilled-checkpoint-path ve --gemma-root, 2.5'te --transformer-path ve --text-encoder-path. Depodaki her pipeline iki sürümle de çalışıyor.

TERMINAL · 2.3 AĞIRLIKLARI

```
# 2.5, bileşen başına ayrı dosya, Gemma ayrı imniyor
hf download Lightricks/LTX-2.5 \
  diffusion_models/ltx-2.5-22b-distilled-transformer-bf16.safetensors \
  text_encoders/gemma-12b-mix-proj-ltx-2.5-bf16.safetensors \
  vae/ltx-2.5-video-vae-bf16.safetensors \
  vae/ltx-2.5-audio-vae-bf16.safetensors \
  latent_upscale_models/ltx-2.5-latent-spatial-upscaler-x2-bf16-1.0.safetensors \
  --local-dir models/ltx-2.5
```

TUZAK

2.3 ve 2.5 dosyaları birbirine karışmaz. İki sürümü aynı klasörde tutma.

UYARI

Beklentini Doğru Kur

Bu 22 milyar parametrelilik bir model ve açık ağırlıklı olması ücretsiz olduğu anlamına gelmiyor. Aşağıdakileri başlamadan önce oku.

LİSANS MIT DEĞİL

Depo LTX-2.x Community License ile geliyor. Yıllık geliri 10 milyon doların altındaki kişi ve şirketler serbestçe kullanabiliyor. Üstündekilerin ticari kullanım için Lightricks'ten ayrı ücretli lisans alması gerekiyor.

BU HER BİLGİSAYARDA DÖNMEZ

22 milyar parametrelilik bir model. Ağırlıklar onlarca gigabayt yer kaplıyor ve tam boyutta çalışmak ciddi VRAM istiyor. FP8 ve offload ile düşürebilirsin ama sihir bekleme.

DEPO 2.3'Ü ARTIK LEGACY SAYIYOR

Bu rehber 2.3'ün tek dosya düzenini anlatıyor, çünkü kurulumu daha sade ve depodaki her pipeline onunla da çalışıyor. Ama README'de 2.3 artık "legacy" başlığı altında, önerilen sürüm 2.5. Bunu bilerek başla.

İKİ SÜRÜMÜN DOSYALARI KARIŞMAZ

2.3 tek dosya, 2.5 bileşen başına ayrı dosya. Bayrak isimleri farklı ve bir LoRA yalnızca eğitildiği sürümle çalışıyor. İki sürümü aynı klasörde tutma.

UZUN VIDEO TEK KOMUTLA ÇIKMIYOR

Model kısa klip üretiyor. Uzun videoyu sahne sahne üretilip birleştiriyorsun. Bunu bilmeden başlarsan hayal kırıklığı olur.

ASIL KAZANÇ ÇÖZÜNÜRLÜK DEĞİL, TEKRAR

Bulut kredisi harcamadığın için "acaba şu açtı daha mı iyi" sorusunu on kere sorabiliyorsun. İyi video tek iyi prompttan değil, ucuz tekrardan çıkıyor.

KAYNAKLAR

Resmî Sayfalar

LTX-2 deposu · Kurulum, komutlar ve pipeline listesi github.com/Lightricks/LTX-2

LTX-2.3 ağırlıkları · Kapılı, şartları kabul etmen gerekiyor huggingface.co/Lightricks/LTX-2.3

Gemma 3 kodlayıcı · 2.3'ün metin kodlayıcısı huggingface.co/google/gemma-3-12b-it-qat-q4_0-unquantized

2.3 model listesi · Tüm 2.3 dosyaları github.com/Lightricks/LTX-2/blob/main/MODELS-LTX-2.5.md

Community License · Ticari kullanım eşliği burada github.com/Lightricks/LTX-2/blob/main/LICENSE-E-2.x

uv · Kurulumda kullanılan paket yöneticisi docs.astral.sh/uv

ffmpeg · Sahneleri tek dosyada birleştirmek için ffmpeg.org



@mirzabuilds · LTX-2 lokal video kiti · yorumla LTX yaz