

# Prompt: video-editing

Sottoinsieme dei prompt del libro filtrati per tema: video-editing Capitoli inclusi: Capitolo 12

---

## Prompt 1

```
sudo apt update && sudo apt install ffmpeg
```

## Prompt 2

```
winget install Gyan.FFmpeg
```

## Prompt 3

```
# Clona il toolkit
git clone https://github.com/digitalsamba/claude-code-video-toolkit.git

# Crea la directory della skill è copia il contenuto
mkdir -p ~/.claude/skills/video-use
cp -r claude-code-video-toolkit/skills/video-use/. ~/.claude/skills/video-use/
```

## Prompt 4

Installa la skill video-use dal Claude Code Video Toolkit su GitHub.  
Controlla che FFmpeg sia installato, poi installa la skill.  
Dimmi quando è pronta.

## Prompt 5

```
/video-use [descrizione dell'operazione che vuoi eseguire]
```

## Prompt 6

Prendi il file conferenza.mp4 nella mia cartella Desktop/video.  
Taglia i primi 3 minuti (intro) e gli ultimi 2 minuti (chiusura).  
Salva il risultato come conferenza-editata.mp4 nella stessa cartella.

## Prompt 7

Taglia il file lezione.mp4 dal minuto 16:30 al minuto 28:45.  
Salva il risultato come lezione-concetto-chiave.mp4  
Usa stream copy (senza ricompressione) per velocità massima.

## Prompt 8

Converti il file tutorial.mp4 in formato verticale 9:16 per Instagram Reel.  
Risoluzione target: 1080x1920.  
Il video è orizzontale: centra il soggetto e riempi il resto con  
uno sfondo sfocato (blur) preso dal video stesso.  
Non usare barre nere.  
Salva come tutorial-reel.mp4

## Prompt 9

Unisci questi file video nell'ordine indicato:

1. intervista-parte1.mp4
2. intervista-parte2.mp4
3. intervista-parte3.mp4

Aggiungi una dissolvenza al nero di 0.5 secondi tra ogni clip.  
Assicurati che l'audio sia continuo e senza salti.  
Salva come intervista-completa.mp4

## Prompt 10

Estrai l'audio dal file intervista.mp4.  
Formato: MP3, qualità 128 kbps, mono (un canale), frequenza 16kHz.  
Salva come audio-per-trascrizione.mp3

Dimmi la dimensione del file risultante.

## Prompt 11

Comprimi il file presentazione.mp4 per ridurlo a meno di 15 MB.  
Mantieni la risoluzione originale se possibile.  
Se la risoluzione deve scendere, massimo 720p.  
Salva come presentazione-compresa.mp4

Dimmi la qualità stimata e la dimensione prima di procedere.

## Prompt 12

Aggiungi il file logo.png come watermark al video tutorial.mp4:  
- Posizione: angolo in basso a destra  
- Margini: 20px dal bordo destro, 20px dal bordo inferiore  
- Opacità: 70% (semitrasparente per non coprire il contenuto)  
- Dimensione logo: 120px di larghezza, mantieni proporzioni  
Salva come tutorial-brandizzato.mp4

## Prompt 13

Crea una GIF animata dal file demo.mp4:  
- Dal minuto 0:45 al minuto 0:58 (13 secondi)  
- Risoluzione 480px di larghezza (standard per GIF)  
- 15 frame al secondo (bilanciamento qualità/dimensione)  
- Ottimizza la palette colori  
Salva come demo-clip.gif e dimmi la dimensione risultante.

## Prompt 14

Aiutami a configurare l'API Whisper di OpenAI passo per passo.  
Non ho mai usato API prima d'ora.  
Parti dall'inizio: dove devo andare, cosa devo cliccare, dove trovare la API key.  
Aspetta la mia conferma a ogni passaggio prima di continuare.

## Prompt 15

Imposta la API key di OpenAI come variabile d'ambiente sul mio sistema.  
La key è: [incolla qui la key, es. sk-proj-...]  
Vuoi che funzioni anche nelle sessioni future, non solo in questa.

## Prompt 16

Verifica che la API key di OpenAI sia configurata correttamente.  
Fai una chiamata di test all'API Whisper con un audio di prova (genera tu il file se non ne ho uno)  
e dimmi se funziona.

## Prompt 17

Aggiungi sottotitoli automatici in italiano al file mio-video.mp4  
usando Whisper API con il modello gpt-4o-transcribe.  
Font Arial Bold, testo bianco con contorno nero 3px, posizione 80% dal basso.  
Prima mostrami il file SRT generato, poi chiedi conferma per procedere con il burn-in.  
Salva il risultato come mio-video-con-sottotitoli.mp4

## Prompt 18

Aiutami a installare Whisper open-source sul mio computer per la trascrizione locale.  
Il mio sistema è [Linux/Mac/Windows con WSL].  
Voglio usarlo per trascrivere video in italiano senza usare l'API a pagamento.

## Prompt 19

Trascrivi il file audio-italiano.mp3 con Whisper locale.  
Usa il modello large-v3 (il migliore per l'italiano).  
Output: file SRT con timestamp.

## Prompt 20

Elabora il file registrazione-raw.mp4 e producilo come Reel Instagram:

1. EDITING: Taglia i primi 40 secondi (intro) e gli ultimi 20 secondi (outro).  
Mantieni tutto il resto.
2. AUDIO: Normalizza il volume a -16 LUFS (standard per social media).
3. FORMATO: Converti in verticale 9:16 (1080x1920).  
Il video è orizzontale 16:9: centra il soggetto e usa sfondo blur (prendi dal video originale).
4. SOTTOTITOLI: Genera i sottotitoli in italiano con Whisper API (gpt-4o-transcribe).  
Mostrami il file SRT e aspetta la mia approvazione prima di procedere.
5. STILE SOTTOTITOLI: Font Arial Bold, 60px, testo bianco, contorno nero 3px.  
Posizione: 15% dal basso (zona sicura da sovrimpressioni Instagram).  
Allineamento: centrato.
6. COMPRESSIONE: Target sotto 100 MB (limite upload Instagram).  
Mantieni la massima qualità possibile entro quel limite.

Salva il risultato finale come reel-pronto.mp4

Prima di iniziare: mostrami il piano completo step-by-step e chiedi conferma.

## Prompt 21

Crea una versione TikTok del file reel-pronto.mp4:

- Formato 9:16, 1080x1920 (stessa risoluzione)
- Durata: se più lungo di 60 secondi, taglia al minuto
- Sottotitoli: stessa posizione ma font 20% più grande (su TikTok si vede più piccolo)
- Aggiungi fade-in nero di 0.5 secondi all'inizio
- Encoding: H.264 High Profile, audio AAC 192kbps

Salva come tiktok-versione.mp4

## Prompt 22

Dove posizionare i sottotitoli per Instagram Reel 1080x1920  
senza che vengano coperti dall'interfaccia Instagram?  
Calcolami la posizione Y ottimale in pixel.

## Prompt 23

Crea una skill "pipeline-social" che esegue automaticamente  
la pipeline completa (taglio → audio → formato → sottotitoli → compressione)  
su qualsiasi video che le passo come argomento.  
Parametri configurabili: formato target (reel/post/tiktok),  
se generare sottotitoli (sì/no), dimensione target in MB.  
Salva la skill in ~/.claude/skills/pipeline-social/

## Prompt 24

/pipeline-social  
Elabora tutti i file .mp4 nella cartella ~/Video/settimana-corrente/  
Formato: reel 9:16  
Sottotitoli: sì, italiano  
Dimensione target: 80 MB  
Salva i risultati in ~/Video/pronti-per-upload/

## Prompt 25

Controlla l'utilizzo dell'API Whisper di questo mese:

- Totale minuti trascritti
- Costo totale
- Proiezione costo fine mese

Vai su [platform.openai.com/usage](https://platform.openai.com/usage) e guida attraverso le sezioni rilevanti.

## Prompt 26

Migliora la qualità audio del file registrazione.mp4 prima della trascrizione:

1. Riduci il rumore di fondo (filtro denoise)
2. Normalizza il volume a -16 LUFS
3. Converti in mono (un canale solo – Whisper lavora meglio)
4. Imposta frequenza di campionamento a 16kHz (ottimale per Whisper)
5. Salva come audio-ottimizzato.wav

Poi usa questo file per la trascrizione invece dell'audio originale.

## Prompt 27

Mostrami il file SRT generato dalla trascrizione di mio-video.mp4

prima di procedere con il burn-in.

Voglio revisionarlo e correggere eventuali errori.

## Prompt 28

Usa il file SRT corretto che ti mostro per il burn-in:

[incolla il contenuto SRT corretto]

Esegui il burn-in con stile standard (Arial Bold 60px, bianco, contorno nero).

## Prompt 29

Trascrivi il file podcast-tech.mp4 in italiano con Whisper API.

Il podcast parla di Claude Code, skill, MCP, subagent, e automazioni AI.

Questi termini appaiono frequentemente e devono essere trascritti correttamente.

Usa il modello gpt-4o-transcribe.

## Prompt 30

/video-use aggiungi sottotitoli da intervista.srt al file intervista.mp4 con questo stile:

Font Arial Black 64px, testo bianco puro (#FFFFFF),

contorno nero 4px, ombra leggera, posizione 82% dal basso, centrato.

Salva come intervista-con-sub-styled.mp4

## Prompt 31

Ho installato FFmpeg ma ottengo "command not found". Aiutami a risolvere.

Il mio sistema è [Linux/Mac/Windows].

## Prompt 32

Il mio file audio è più grande di 25 MB e Whisper API lo rifiuta.

Dividilo in segmenti da 5 minuti, invia ogni segmento a Whisper separatamente,

e poi unisci i file SRT risultanti in ordine corretto.

Il file si chiama audio-lungo.mp3

## Prompt 33

I sottotitoli nel video edited-final.mp4 sono desincronizzati:

arrivano circa 2 secondi in ritardo rispetto all'audio.

Correggi il file SRT spostando tutti i timestamp di -2 secondi.

## Prompt 34

La trascrizione Whisper del file riunione.mp4 ha troppi errori –

audio di qualità bassa, microfono distante, rumore di fondo.

Applica prima il filtro denoise aggressivo di FFmpeg e riprova.

Usa l'opzione afftdn per la riduzione rumore.

## Prompt 35

Il file video-senza-audio.mp4 non ha traccia audio.

Ho una registrazione audio separata in audio-registrazione.wav.

Sincronizzali: l'audio inizia 3 secondi dopo l'inizio del video.

Salva come video-sincronizzato.mp4

## Prompt 36

I sottotitoli in video.mp4 spesso escono dai bordi laterali.

Riapplica i sottotitoli con queste limitazioni:

- Massimo 35 caratteri per riga
- A capo automatico se la riga è più lunga
- Margini laterali di 50px da ogni bordo