

ذخیره ویدیو:

بر خلاف کار با عکس ها، کار با ویدیو از پیچیدگی های بیشتری برخوردار است و صرفاً با یک دستور خاص نمی توان یک فایل ویدیویی را ذخیره نمود.

در هنگام کار با ویدیو، مفاهیم **کدک (Codec)** و **FourCC** نقش مهمی در نحوه ذخیره سازی، فشرده سازی و پخش ویدیو ایفا می کنند. در اینجا به توضیح هر یک می پردازیم:

1 کدک: (Codec)

واژه "کدک" مخفف عبارت "**Coder-Decoder**" (رمزگذار-گشاینده) است. کدک در واقع یک الگوریتم نرم افزاری یا سخت افزاری است که برای انجام دو وظیفه اصلی استفاده می شود:

- **رمزگذاری (Encoding):** فرآیند فشرده سازی داده های ویدیویی (و احتمالاً صوتی) به یک فرمت کارآمدتر برای ذخیره سازی یا انتقال. هدف از رمزگذاری کاهش حجم فایل ویدیو بدون از دست دادن کیفیت قابل توجه است.
- **گشایش (Decoding):** فرآیند برگرداندن داده های فشرده شده به فرمت اصلی برای پخش یا ویرایش.

چرا به کدک نیاز داریم؟

ویدیوهای خام حجم بسیار زیادی دارند. برای مثال، یک دقیقه ویدیوی HD غیرفشرده می تواند چندین گیگابایت فضا اشغال کند. استفاده از کدک ها این امکان را فراهم می کند تا:

- فضای ذخیره سازی کمتری برای ذخیره ویدیوها مورد نیاز باشد.
- انتقال ویدیوها از طریق اینترنت یا شبکه های دیگر سریع تر و کارآمدتر انجام شود.
- پخش ویدیوها بر روی دستگاه های مختلف با منابع محدود امکان پذیر شود.

انواع کدک ها:

کدک های ویدیویی متنوعی وجود دارند که هر کدام از الگوریتم های فشرده سازی متفاوتی استفاده می کنند و ویژگی ها و کیفیت های مختلفی دارند. برخی از کدک های رایج عبارتند از:

- **H.264 (AVC)** یکی از پرکاربردترین کدک ها که تعادل خوبی بین کیفیت و فشرده سازی ارائه می دهد و تقریباً توسط تمام دستگاه ها و مرورگرها پشتیبانی می شود.

- **H.265 (HEVC)** نسل جدیدتر H.264 که فشرده‌سازی بهتری را با حفظ کیفیت مشابه ارائه می‌دهد، اما ممکن است به منابع بیشتری برای رمزگذاری و گشایش نیاز داشته باشد و پشتیبانی آن هنوز به گستردگی H.264 نیست.
- **VP9** یک کدک رایگان و بازمتن که توسط گوگل توسعه یافته و به طور گسترده در وب استفاده می‌شود.
- **AV1** یک کدک جدیدتر، رایگان و بازمتن که هدف آن ارائه فشرده‌سازی بهتر از H.265 است، اما هنوز در مراحل پذیرش اولیه قرار دارد.
- **MPEG-4** یک استاندارد قدیمی‌تر که شامل چندین کدک مختلف می‌شود.
- **Xvid و DivX** کدک‌های محبوب برای فشرده‌سازی ویدیوهای خانگی.

FourCC (Four-Character Code):

FourCC یک شناسه چهار بایتی (چهار کاراکتری) است که برای تعیین نوع کدک مورد استفاده در یک فایل ویدیویی به کار می‌رود. این کد معمولاً به صورت یک رشته چهار کاراکتری ASCII نمایش داده می‌شود.

کاربرد: FourCC

- **شناسایی کدک**: نرم‌افزارهای پخش و ویرایش ویدیو از کد FourCC برای تشخیص اینکه یک فایل ویدیویی با کدام کدک رمزگذاری شده است، استفاده می‌کنند. این اطلاعات به نرم‌افزار کمک می‌کند تا بداند چگونه باید ویدیو را گشایش و پخش کند.
- **سازگاری**: وجود کد FourCC به سیستم عامل و نرم‌افزارها کمک می‌کند تا بررسی کنند آیا کدک مورد نیاز برای پخش یک فایل ویدیویی بر روی سیستم نصب شده است یا خیر. در صورت عدم وجود کدک مناسب، ممکن است پخش ویدیو با مشکل مواجه شود یا اصلاً امکان‌پذیر نباشد.
- **فرمت‌های فایل**: کد FourCC معمولاً در ساختار فایل‌های ویدیویی مانند AVI، MP4 و MOV ذخیره می‌شود.

مثال‌های FourCC

برخی از کدهای FourCC رایج عبارتند از:

- DIVX برای کدک DivX
- XVID برای کدک Xvid
- H264 یا AVC1 برای کدک H.264
- HEVC یا hvc1 برای کدک H.265
- VP90 برای کدک VP9

• MP4V برای کدک MPEG-4 Part 2

در: OpenCV

همانطور که در جزوه قبلی اشاره شد، متد `cap.get(cv2.CAP_PROP_FOURCC)` در **OpenCV** برای دریافت کد **FourCC** مربوط به ویدیویی که توسط شیء **VideoCapture** باز شده است، استفاده می‌شود. مقدار برگشتی یک عدد صحیح است که می‌توانید آن را با استفاده از تابع `cv2.VideoWriter_fourcc(*'XVID')` به یک کد **FourCC** تبدیل کنید یا آن را به صورت کاراکتری تفسیر کنید تا کدک مورد استفاده را شناسایی کنید.

رابطه بین کدک و FourCC

کدک الگوریتم واقعی فشرده‌سازی و گشایش ویدیو است، در حالی که **FourCC** یک شناسه کوتاه برای آن کدک است. فایل ویدیویی حاوی داده‌های فشرده‌شده توسط یک کدک خاص است و کد **FourCC** موجود در فایل به نرم‌افزارها می‌گوید که از کدام کدک برای پخش آن استفاده کنند.

به طور خلاصه، کدک‌ها مغز متفکر فشرده‌سازی و گشایش ویدیو هستند و **FourCC** یک برچسب شناسایی است که به سیستم و نرم‌افزارها کمک می‌کند تا بدانند با کدام کدک سر و کار دارند. درک این مفاهیم برای کار با ویدیو در محیط‌های مختلف، از جمله **OpenCV**، ضروری است.

حال که با اصطلاحات ضروری آشنا شدید ببینیم که چگونه می‌توانیم یک ویدیو دریافت شده از وبکم را ذخیره کنیم:

ابتدا با استفاده از خط زیر یک کدک برای ویدیو ایجاد میکنیم:

```
fourcc = cv2.VideoWriter_fourcc(*'XVID')
```

cv2.VideoWriter_fourcc() این یک تابع از ماژول **cv2** است که برای تعیین کدک (codec) مورد استفاده برای فشرده‌سازی ویدیو استفاده می‌شود. کدک‌ها الگوریتم‌هایی هستند که برای رمزگذاری و رمزگشایی داده‌های ویدئویی به منظور کاهش حجم فایل و حفظ کیفیت استفاده می‌شوند.

'XVID' * این قسمت یک رشته **'XVID'** را به عنوان آرگومان به تابع `cv2.VideoWriter_fourcc()` می‌دهد. عملگر ستاره (*) قبل از رشته، کاراکترهای آن را از هم جدا کرده و به عنوان آرگومان‌های جداگانه به تابع ارسال می‌کند. بنابراین، این معادل فراخوانی `cv2.VideoWriter_fourcc('X', 'V', 'I', 'D')` است.

'XVID': این یک کد چهار کاراکتری (FourCC) است که نشان‌دهنده کدک MPEG-4 Part 2 است. این یک کدک ویدیویی محبوب و نسبتاً کارآمد است که تعادل خوبی بین کیفیت و اندازه فایل ارائه می‌دهد.

fourcc = ...: نتیجه فراخوانی تابع `cv2.VideoWriter_fourcc(*'XVID')` که کدک XVID را مشخص می‌کند (در متغیری به نام `fourcc` ذخیره می‌شود. این متغیر بعداً در هنگام ایجاد شیء `VideoWriter` برای تعیین کدک مورد استفاده قرار خواهد گرفت.

بعد از ایجاد کدک مورد نظر می‌توانیم فایل ویدیویی مورد نظر را با استفاده از دستور زیر ذخیره نماییم:

```
output = cv2.VideoWriter('output.avi', fourcc, 30.0, (1280,720))
```

cv2.VideoWriter(): این تابع از ماژول `cv2` برای ایجاد یک شیء `VideoWriter` استفاده می‌شود. این شیء مسئول نوشتن فریم‌های ویدئویی در یک فایل ویدیویی است.

'output.avi': این اولین آرگومان تابع است و نام فایلی را مشخص می‌کند که ویدیو در آن ذخیره خواهد شد. در این مورد، نام فایل `'output.avi'` است و پسوند `.avi` نشان می‌دهد که ویدیو در قالب AVI (Audio Video Interleave) ذخیره می‌شود.

fourcc: این دومین آرگومان است و کدک مورد استفاده برای فشرده‌سازی ویدیو را مشخص می‌کند. همانطور که در خط اول توضیح داده شد، این متغیر حاوی کدک XVID است.

30.0: این سومین آرگومان است و نرخ فریم (frames per second - FPS) ویدیو را تعیین می‌کند. در اینجا، نرخ فریم روی 30 فریم در ثانیه تنظیم شده است، به این معنی که 30 تصویر ثابت در هر ثانیه از ویدیو نمایش داده خواهند شد.

(1280, 720): این چهارمین آرگومان یک تاپل است که ابعاد (عرض و ارتفاع) فریم‌های ویدیو را در پیکسل مشخص می‌کند. در این مورد، ویدیو با رزولوشن 1280 پیکسل در عرض و 720 پیکسل در ارتفاع (که به عنوان HD یا 720p نیز شناخته می‌شود) ذخیره خواهد شد.

output = ...: نتیجه فراخوانی تابع `cv2.VideoWriter()` که یک شیء `VideoWriter` پیکربندی شده برای ذخیره ویدیو با مشخصات داده شده است (در متغیری به نام `output` ذخیره می‌شود. از این شیء `output` بعداً برای نوشتن فریم‌های ویدیویی واقعی در فایل `'output.avi'` استفاده خواهد شد.

insta: kahkeshani_mohammad

دوره پردازش تصویر و بینایی کامپیوتر با open cv

youtube: <https://www.youtube.com/@mohammadkahkeshani>

• مدرس محمد کهکشانی (مدرس رسمی دانشگاه هاروارد)

به طور خلاصه، خط دوم یک شیء `VideoWriter` ایجاد می‌کند که پیکربندی شده است تا ویدیویی با نام `'output.avi'` با استفاده از کدک `XVID`، با نرخ فریم 30 فریم در ثانیه و با رزولوشن `1280*720` ذخیره کند. این شیء در متغیر `output` نگهداری می‌شود و برای نوشتن فریم‌های ویدیو مورد استفاده قرار می‌گیرد.