

دانشکده مهندسی کامپیوتر

ارائه مطالب علمی و فنی

مستند پروژه

---

# بررسی الگوریتم‌های فشرده‌سازی و کاربردهای آنها

---

نگارندگان:  
محمد مهدی عرفانیان

۱۸ آذر ۱۳۹۸



## چکیده

در مستندی که پیش روی خواننده عزیز قرار دارد تلاش شده تا مختصراً الگوریتم‌های فشرده‌سازی مختلف و کاربردهای آن‌ها در زمینه‌های مختلف مهندسی کامپیوتر در راستای انجام پروژه درس ارائه مطالب علمی و فنی بررسی شود. این درس در پاییز ۹۸ در دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف توسط دکتر همت‌یار ارائه شده است.

برای سهولت کار استاد محترم درس برای تحصیل اطمینان از درستی مستندسازی و همچنین استفاده دانش‌جویان علاقه‌مند، سیر پیشرفت مستند به همراه کدهای  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  در *Github* قرار گرفته است، لازم به ذکر است که این مستند به صورت متن‌باز ارائه شده و استفاده از آن بدون ذکر منبع برای همگان آزاد است. در انتها از استاد محترم درس، دستیار آموزشی ایشان و خوانندگان محترم تشکر می‌کنم.

با آرزوی خوش‌وقتی برای تمامی خوانندگان این مستند  
نگارنده

# فهرست مطالب

۴	۱	مقدمه
۵	۱.۱	تعریف
۵	۲.۱	انواع الگوریتم‌های فشرده‌سازی
۵	۱.۲.۱	الگوریتم‌های Lossless
۶	۲.۲.۱	الگوریتم‌های Lossy
۸	۲	بررسی نحوه فشرده‌سازی در فرمت JPEG
۹	۱.۲	DCT
۹	۱.۱.۲	تعریف
۹	۲.۲	تبدیل دنباله‌ای از پیکسل‌ها به تابع
۹	۳.۲	نحوه کار JPEG

# فهرست تصاویر

۱.۱	نمونه فایل تولیدشده توسط نگارنده برای تست میزان کمپرس در فرمت PNG . . . . .	۶
۱.۲	توصیف فرکانسی توابع $\cos(x), \cos(2x)$ . . . . .	۱۰
۲.۲	توصیف فرکانسی تابع $\cos(x) + \cos(2x)$ . . . . .	۱۰

# فهرست جداول

۶	.....	حجم فایل نمونه در فرمت PNG	۱.۱
۷	.....	حجم فایل نمونه در فرمت JPG	۲.۱

# فصل ۱

## مقدمه

توضیحی اولیه مبنی بر تعریف کلی فشرده‌سازی، انواع الگوریتم‌ها و کاربردها

## ۱.۱ تعریف

الگوریتم‌های فشرده‌سازی، الگوریتم‌هایی هستند که می‌توان با استفاده از آن‌ها داده‌ها را طوری رمزنگاری کرد که در تعداد کمتری بیت نسبت به آرایش اولیه قابل ارائه باشند. برای مثال می‌دانیم که برای ذخیره هر بیت اسکی هشت بیت فضا لازم است، می‌توان با استفاده از نگاشتی متشکل از حروف استفاده‌شده در یک متن تعداد بیت‌های مورد نیاز برای نشان دادن هر حرف استفاده‌شده در متن را کاهش داد. با استفاده از این تکنیک در حقیقت متن را در قالب جدیدی فشرده کرده‌ایم.

## ۲.۱ انواع الگوریتم‌های فشرده‌سازی

در یک دسته‌بندی الگوریتم‌های فشرده‌سازی را به دو نوع زیر افراز می‌کنند.

- Lossless یا بدون هدررفت داده
- Lossy یا همراه هدررفت داده

### ۱.۲.۱ الگوریتم‌های Lossless

در این سری الگوریتم‌ها داده ورودی بدون هیچ‌گونه هدررفتی از داده خروجی قابل بازیابی است، الگوریتم‌های این دسته با استفاده از افزودن آمارهای تلاش می‌کنند تا نحوه نمایش داده را در نگاشتی به نحوه نمایش دیگری که به فضای کمتری نیاز دارد تبدیل کنند. این الگوریتم‌ها در مواقعی که ثابت ماندن داده در طی فشرده‌سازی الزامی است استفاده می‌شوند، همچنین معمولاً برای بازیابی اطلاعات فشرده‌شده نیاز به داده‌هایی خارجی است که با کمک آن عمل بازیابی انجام می‌گیرد، از این رو می‌توان از این نوع الگوریتم‌ها در رمزنگاری نیز استفاده کرد. یکی از کاربردهای اصلی این الگوریتم‌ها در فشرده‌سازی متون است که اشتباه شدن حتی یک حرف می‌تواند باعث بدخوانی و بدفهمی متن اصلی گردد. الگوریتم‌های مشهور کمپرس Lossless به شرح زیر اند.

- Run-Length Encoding (RLE)
- Lempel-Ziv (LZ)
- Huffman Encoding
- Burrows Wheeler Transform

البته لازم به ذکر است که در عمل از مجموعه‌ای از الگوریتم‌های فوق برای رسیدن به درصد مطلوب فشرده‌سازی استفاده می‌شود.

### نمونه‌های الگوریتم‌های Lossless

در عمل از الگوریتم‌های Lossless در مواقعی که پایداری داده‌های ذخیره‌شده حیاتی است یا این که فایل در آینده به تعداد زیادی بار فشرده و گسترده می‌شود و از دست دادن قسمتی از داده در هربار فشرده‌سازی منجر به اختلاف فاحش نهایی خواهد شد استفاده می‌شود.

#### • PNG

در طراحی فرمت PNG برای فشرده‌سازی تصاویر از الگوریتم Lempel-Ziv-Welch (LZW) که الگوریتمی Lossless است استفاده شده. در شکل ۱.۱ و جدول ۱.۱ یک نمونه عکس در حالت فشرده‌نشده و فشرده‌شده با فرمت PNG و مقدار حجم آن در حالت‌های مختلف آورده شده است.

```

۸ ۱ ۸ ۵ ۱ ۰ ۰ ۱ ۱ ۸
۸ ۴ ۴ ۳ ۹ ۱ ۷ ۳ ۰ ۵
۸ ۹ ۷ ۰ ۵ ۲ ۸ ۹ ۲ ۰
۴ ۱ ۶ ۰ ۱ ۰ ۲ ۸ ۶ ۳
۱ ۴ ۷ ۲ ۱ ۵ ۴ ۷ ۵ ۶
۷ ۱ ۲ ۳ ۶ ۷ ۵ ۵ ۹ ۰
۴ ۸ ۲ ۱ ۲ ۲ ۱ ۶ ۲ ۶
۶ ۰ ۵ ۹ ۵ ۴ ۹ ۹ ۶ ۱
۲ ۳ ۹ ۹ ۵ ۴ ۵ ۴ ۰ ۹
۴ ۴ ۴ ۸ ۶ ۶ ۰ ۹ ۹ ۴

```

شکل ۱.۱: نمونه فایل تولیدشده توسط نگارنده برای تست میزان کمپرس در فرمت PNG

جدول ۱.۱: حجم فایل نمونه در فرمت PNG

Format	Size
BMP	۷.۷ مگابایت
PNG	۹۸ کیلوبایت

#### • Free Lossless Audion Codec (FLAC)

الگوریتمی که با استفاده از اطلاعات ذاتی داده‌های صوتی به فشرده‌سازی آنها می‌پردازد، نرخ فشرده‌سازی این الگوریتم با توجه به سطح فشرده‌سازی سازی آن معمولاً بین ۴۰ تا ۶۰ درصد می‌باشد اما در حالت بیشینه ممکن است تا ۸۰ درصد هم برسد.

### ۲.۲.۱ الگوریتم‌های Lossy

در این الگوریتم‌ها پس از هر بار فشرده‌سازی مقداری از داده‌ها از دست می‌روند، معیار ارزیابی این الگوریتم‌ها مقدار فشرده‌سازی با توجه به میزان هدررفت داده می‌باشد، به علت هدررفت مقداری از داده این الگوریتم‌ها معمولاً در مواردی که هدررفت اندک داده توسط انسان یا ماشین قابل تشخیص نباشد استفاده می‌شوند، مثلاً تکنیک‌های ذخیره‌سازی تصاویر و ویدئوها در کامپیوترها مبتنی بر الگوریتم‌های Lossy است زیرا چشم انسان قادر به تشخیص عوض شدن تعدادی پیکسل در صفحه پس از بازیابی فایل فشرده‌شده نیست.

معمولاً در فشرده‌سازی Lossy از Transform Coding استفاده می‌شود که داده‌ها را از فضای حقیقی به فضایی دیگر (معمولاً فرکانس) می‌برد و در آنجا از قسمت‌هایی از داده که تاثیرگذاری و حس‌پذیری کمتری نسبت به دیگران دارند صرف نظر می‌شود، سپس وارون تبدیل اجرا شده و داده‌های کوچک‌شده جدید با الگوریتم‌های Lossless فشرده می‌شوند. یکی از مشهورترین Transform Coding ها الگوریتم Discrete Cosine Transform (DCT) است که در فصول آتی این مستند بیشتر با آن آشنا خواهیم شد.

#### نمونه‌های الگوریتم‌های Lossy

الگوریتم‌های Lossy با همه‌گیر شدن اینترنت و اشتراک‌گذاری بیشتر مدیا در فضای اینترنت بسیار همه‌گیر شدند، از این رو اکثر این الگوریتم‌ها برای فشرده‌سازی فایل‌های صوتی-تصویری یا به اصطلاح Media



استفاده می‌شوند.

- JPEG
- MP3
- MP4
- H.26x

به عنوان نمونه برای الگوریتم Lossy حالت فشرده‌شده عکس ۱.۱ با فرمت JPEG با کیفیت‌های مختلف در جدول ۲.۱ آورده شده است.<sup>۱</sup>

جدول ۲.۱: حجم فایل نمونه در فرمت JPG

Format	Size	Quality(%)
BMP	۷.۷ مگابایت	۱۰۰
PNG	۹۸ کیلوبایت	۱۰۰
JPG	۸.۹۶ کیلوبایت	۹۰
JPG	۹.۶۲ کیلوبایت	۵۰
JPG	۴۹ کیلوبایت	۲۰

<sup>۱</sup> در صورتی که تمایل به دیدن اصل فایل‌ها دارید می‌توانید به *Github* مستند مراجعه کنید.

## فصل ۲

# بررسی نحوه فشرده‌سازی در فرمت JPEG

توضیح نحوه کار فشرده‌سازی در فرمت *JPEG*، مقدمه‌ای بر *DCT* و توضیح کلی *Huffman encoding*

## ۱.۲ DCT

### ۱.۱.۲ تعریف

تبدیل کسینوسی گسسته (به انگلیسی: Discrete cosine transform)، دنباله ای محدود از اعداد (داده ها) را به صورت مجموع توابع کسینوسی با فرکانس های متفاوت نمایش می دهد. تبدیل کسینوسی گسسته، شباهت بسیاری به تبدیل فوریه گسسته (DFT) دارد، با این تفاوت که حاصل تبدیل فقط مقادیر حقیقی دارد (بر خلاف تبدیل فوریه که منجر به مقادیر مختلط می شود).  
به صورت علمی تر می توان نوشت که DCT تابعی معکوس پذیر و خطی از  $R^N$  به  $R^N$  است. فرمول کلی DCT برای فضای یک بعدی به شکل زیر است.

$$X_k = \sum_{n=0}^{N-1} x_n \cos[\pi/n(n+1/2)k], k = 0, \dots, N-1 \quad (1.2)$$

## ۲.۲ تبدیل دنباله ای از پیکسل ها به تابع

هر تصویر در حقیقت به صورت چهار ماتریس دوبعدی ذخیره می شود که هر ماتریس مقدار رنگی پیکسل را برای آن کانال رنگی (RGB) و  $\alpha$  نشان می دهد، برای کمپرس کردن یک عکس ماتریس های کانال های رنگی مختلف را جداگانه فشرده می کنیم، کانال رنگی R را در نظر بگیریم، در این حالت یک ماتریس  $n*m$  داریم که هر خانه آن عددی بین ۰ تا ۲۵۵ را نشان می دهد، برای سادگی یک سطر از این ماتریس را در نظر بگیریم، این سطر معادل با یک آرایه از اعداد می باشد، در صورتی که تمامی اعداد را از دامنه ۰ تا ۲۵۵ به دامنه ۱۲۸ - ۱۲۷ ببریم می توانیم این دنباله را با مجموعه ای از توابع کسینوسی بازتولید کنیم، این ضرایب همان ضرایبی ست که با استفاده از DCT می خواهیم به دست بیاوریم.

شکل ۱.۲ نمونه ای از توصیف عکس با توابع کسینوسی را نشان می دهد.

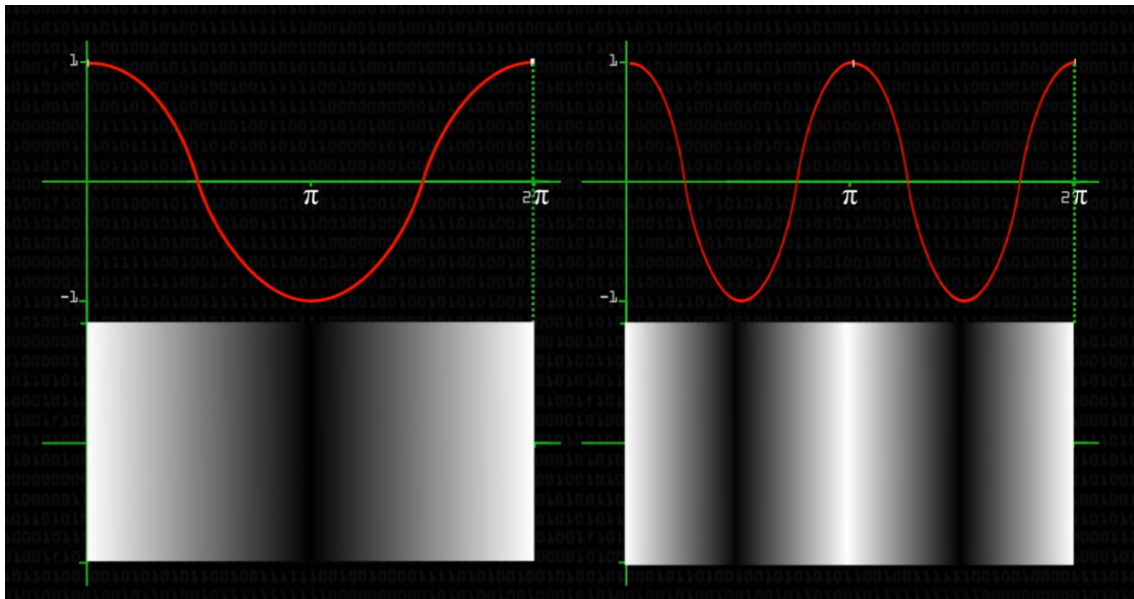
حال اگر دو تابع نشان داده شده در شکل ۱.۲

را با هم جمع کرده و میانگین بگیریم، به نمایش فرکانس دیگری می رسیم که در شکل ۲.۲ نشان داده شده است. در صورتی که به شکل های متفاوت و با ضرایب متفاوت توابع مختلف کسینوسی را با هم جمع کنیم می توانیم هر فرکانسی را بسازیم. این کاری ست که در عمل الگوریتم DCT برای ما انجام می دهد.

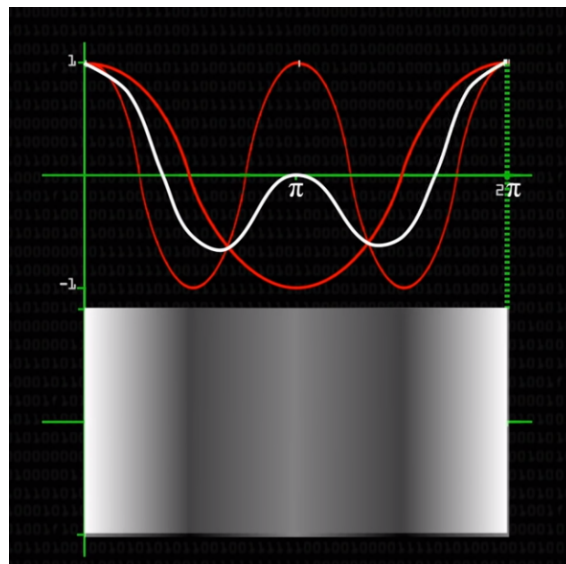
## ۳.۲ نحوه کار JPEG

برای فشرده سازی یک تصویر در JPEG مراحل زیر انجام می شود.

- تبدیل تصویر به بلوک های کوچک
- تبدیل هر بلوک به مجموعه ای از توابع کسینوسی استاندارد با DCT
- تنظیم ضرایب متناسب برای بلوک های DCT
- کمپرس کردن ضرایب با استفاده از Huffman



شکل ۱.۲: توصیف فرکانسی توابع  $\cos(x)$ ,  $\cos(2x)$



شکل ۲.۲: توصیف فرکانسی تابع  $\cos(x) + \cos(2x)$