فهرست مطالب

مقدمه ناشر ۶

مقدمه مؤلف ۷

فصل اول

پیش‌پردازش ۸

تصاویر و عکس‌ها ۸
تصویر در دوربین ۸
مفهوم پردازش تصویر و کاربردهای آن ۹
مفهوم پیکسل و عمق بیت در تصویر ۱۰
مفهوم بعد در یک تصویر ۱۲
برداشت اصولی تصویر ۱۲
نحوهٔ ورود تصویر به رایانه ۱۳

فصل دوم

پردازش تصویر ۱۴

جعبه ابزار پردازش تصویر ۱۴
عکس در MATLAB ۱۴
درک فضاهای رنگی و تبدیل فضای رنگی ۱۴
انواع مدل‌های رنگی ۱۵
انواع داده‌های عددی در MATLAB ۱۷
انواع تصاویر رقومی ۱۸
اصول فراخوانی تصاویر در نرم‌افزار MATLAB ۲۱
ارسال و بازنویسی عکس ۲۵
Colormap ۲۸
نمایش تصاویر گرافیکی در MATLAB ۲۹
تبدیلات تصاویر ۳۱
تبدیل سایر عکس‌ها به باینری ۴۰
تبدیل فضاهای رنگی ۴۲

فصل سوم

نمایش تصاویر به‌صورت همزمان ۴۵

ترسیم چندین عکس در یک Figure ۴۵
نمایش تصویر با imview ۴۸
Subimage ۴۹
گرفتن اطلاعات عکس ۴۹

فصل چهارم

ویرایش تصویر ۵۱

بزرگنمایی تصاویر ۵۱
مفهوم texture Mapping ۵۱
انواع پسوند تصاویر ۵۲
فرمت‌های قابل پشتیبانی در ۵۲
اصول تبدیل کلاس‌ها ۵۵
توابع تبدیل ۵۶
از دست دادن اطلاعات در تبدیلات عکس ۵۹
مفهوم هیستوگرام تصاویر ۵۹
برش نواحی دلخواه از تصاویر ۶۲
تبدیلات هندسی ۶۳
تصحیح تصاویر برای حجم کمتر در حافظه ۶۶
گرفتن اطلاعات پیکسل‌ها ۶۸

فصل پنجم

آرایه چند فریمی ۷۳

پردازش آرایه‌های عکس چند فریمی ۷۳
Image sequences ۷۳
Montage ۷۳
روش‌های ساخت عکس چند فریمی ۷۵
ساخت فیلم از عکس چند فریمی ۷۷
نحوهٔ اتصال پیکس‌ها و پیداکردن اجزای اتصال ۷۹
تبدیل ماتریس برچسب به عکس رنگی ۸۴
اندازه‌گیری خصوصیات اجزای اتصال ۸۵

فصل ششم

بهینه‌سازی تصویر ۹۱

الگوریتم‌های شناسایی لبه ۹۱
لبه ۹۱
مفهوم نویز ۹۳
تعادل در تصویر ۹۵
اصول فیلترکردن ۹۹
عملیات همسایگی کشویی Sliding neighborhood ۹۹
اجرای یک عملیات sliding neighborhood ۱۰۱
طراحی فیلتر خطی و غیرخطی ۱۰۶
تبدیل ماتریس به عکس ۱۲۰
پردازش یک ناحیه از تصویر ۱۲۲
Morphological opening ۱۲۷
تجزیه عنصر ساختاری ۱۳۱
گسترش و فرسایش مورفولوژی ۱۳۲
انجام چهار عمل روی عکس ۱۴۴
اصلاح شدت روشنایی و بررسی اشیاء پس‌زمینه ۱۴۹

مقدمه ناشر

مقدمه مؤلف

پردازش تصویر شاخه ای از علم پردازش سیگنال است که امروزه مهندسان متعددی برای انجام امور پژوهشی و میدانی خود نیازمند یادگیری آن هستند. نرم افزار های زیادی در این زمینه مورد استفاده قرار می گیرد. MATLAB یکی از نرم افزارهایی است که می توان از آن در پردازش تصویر استفاده کرد. در این نرم افزار جعبه ابزاری وجود دارد که قابلیت آنالیز، تجسم، بهبود عکس، رفع تاری، کاهش نویز و . . . را دارا است. از کاربرد پردازش تصاویر میتوان به استفاده پزشکی، صنعت، کشاورزی، نظامی و امنیتی و. . . اشاره کرد.

این کتاب بر اساس استاندارد پردازش و بهینه سازی تصویر با نرم افزار MATLAB سازمان آموزش فنی و حرفه ای کشور به شماره ۱-۰۲۰-۰۵-۲۱۵۱ تهیه و تدوین شده است.

در پایان از همه دوستان و عزیزانی که در مدت زمان تالیف و تدوین این کتاب، از همفکری و همکاری آن ها برخوردار بوده ایم صمیمانه قدردانی و تشکر می نمائیم.

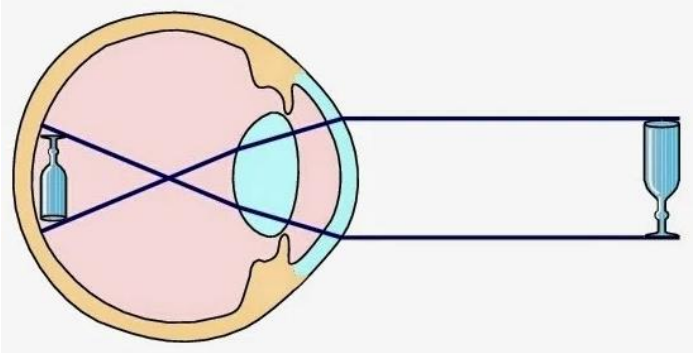
فصل اول

پیش‌پردازش

تصاویر و عکس‌ها

قبل از بحث در مورد شکل‌گیری و تشکیل تصویر در دوربین بهتر است که در خصوص نحوهٔ شکل‌گیری تصویر در چشم انسان صحبت شود. اصول کار دوربین بر پایهٔ عملکرد چشم انسان است.

نور به جسم برخورد می‌کند، پس از برخورد به چشم بازتاب می‌شود. نور بازتاب‌شده از چشم عبور می‌کند. ضمن عبور از چشم زاویه خاصی را تشکیل می‌دهد. تصویر بر روی شبکیه که سمت عقب دیواره است تشکیل می‌شود. این تصویر به‌صورت وارونه است. مغز انسان این تصویر را تفسیر می‌کند و سبب می‌شود اشیاء را آن‌طور که هستند، دیده و درک شوند. علت تشکیل زاویه نور در چشم این است که انسان را قادر سازد عمق و ارتفاع را درک کند. شکل ۱-۱ نحوهٔ تشکیل تصویر در چشم را نشان می‌دهد.



شکل ۱-۱

تصویر در دوربین

آنالوگ

در این نوع دوربین‌ها تشکیل تصویر به‌دلیل واکنش شیمیایی بر روی نوار است. نور از طریق لنز دوربین عبور می‌کند و بر روی نوارهای حاوی ذرات هالید نقره قرار می‌گیرد و یک واکنش صورت می‌گیرد و سبب می‌شود که یک تصویر به‌صورت منفی از آن تشکیل شود.

دیجیتال

ساختار دوربین‌های دیجیتال مانند دوربین آنالوگ است، ولی در دوربین‌های دیجیتال شکل‌گیری تصویر به‌دلیل واکنش شیمیایی نیست، بلکه به‌دلیل وجود قطعه‌ای حساس به نور به نام سنسور است. این قطعه CCD نام دارد و مخفف عبارت Charge Coupled Device است. CCD یک عنصر نیمه هادی است و تصاویر را به سیگنال‌های دیجیتال تبدیل می‌کند و از قطعات کوچکی مانند یک شبکه ایجاد شده است. به عبارت دیگر CCD معمولاً به شکل مستطیل و شبیه یک ماتریس است که هر عنصر آن یک سنسور است.

هنگام عکس‌گرفتن با این نوع دوربین، نوری که از جسم منعکس می‌شود، از طریق لنز وارد آن می‌شود. تصویر بر روی CCD تشکیل می‌شود. هنگامی که یک پیکسل روی CCD نور را دریافت می‌کند، یک بار الکتریکی متناسب با شدت نور تولید می‌شود. بار الکتریکی به سیگنال الکتریکی تبدیل می‌شود تا شدت نور دریافت شده توسط هر پیکسل به‌دست آید. این بدین معنی است که هر پیکسل یک سنسور است که می‌تواند شدت نور را تشخیص دهد. یک CCD با ۲ میلیون پیکسل مجموعه‌ای از ۲ میلیون دیود نوری و حسگر است.

مفهوم پردازش تصویر و کاربردهای آن

پردازش تصویر

پردازش تصویر روشی است برای پیاده‌سازی برخی عملیات بر روی عکس که سبب بهینه‌سازی و یا استخراج اطلاعات مفید از آن می‌شود. این عملیات نوعی پردازش سیگنال است که به یک تصویر به‌عنوان ورودی اعمال می‌شود و نتیجه آن در تصویر خروجی نمایش داده می‌شود. سه مرحله کلی که همه انواع داده‌ها باید با استفاده از تکنیک پردازش تصویر طی کنند، پیش‌پردازش، بهبود، نمایش و استخراج اطلاعات است.

کاربردهای علم پردازش

- ✓ **کاربرد صنعتی:** کنترل کیفت.
- ✓ **کاربرد پزشکی:** استفاده در MRI, CT-SCAN, X-Ray.
- ✓ **کاربرد نظامی:** تشخیص و هدف‌یابی خودکار.
- ✓ **کاربرد امنیتی:** تشخیص چهره، کنترل سرعت.

پردازش تصویر به‌صورت دیجیتال، نمایشی از یک تصویر در فضای دوبعدی به‌عنوان مجموعه و گستره‌ای محدود از مقادیر دیجیتال است که عناصر یا پیکسل‌های تصویر نامیده می‌شوند.

مفهوم پیکسل و عمق بیت در تصویر

pixel

pixel از عبارت Picture Element گرفته شده است. پیکسل کوچکترین عنصر یک تصویر است که می‌تواند دارای یک مقدار باشد. در یک عکس سیاه و سفید این مقدار می‌تواند ۰ و یا ۱ باشد. مقدار پیکسل در هر نقطه از تصویر به شدت دریافت نور برخورد شده به آن بستگی دارد. به عبارت دیگر رنگ هر پیکسل مقداری مناسب با شدت نور برخورد شده در آن نقطه خاص است.

برای مثال، عکس شکل ۱-۲ را درنظر بگیرید.

شکل ۲-۱

روی تصویر زوم کنید تا مربعات تشکیل‌دهندهٔ عکس ظاهر شود. این مربعات همان پیکسل است.

شکل ۱-۳

عمق بیت

بیت کوچکترین واحد قابل پردازش در رایانه است. مقدار هر پیکسل در یک یا چند بیت ذخیره می‌شود. تعداد بیت‌ها در هر پیکسل را عمق بیت می‌نامند. عمق بیت، وضوح تصویر را تعیین می‌کند. این خصوصیت در نمایشگرها با ppi (pixel per inch) و در چاپگرها dpi (dot per inch) اندازه‌گیری می‌شود.

پیکسل هر عکس می‌تواند دارای مقدار عددی متفاوتی باشد. در یک عکس سیاه و سفید که دارای دو رنگ است، پیکسل‌ها می‌توانند دارای مقادیر یک و یا صفر باشند. صفر برای رنگ سیاه و یک برای رنگ سفید است. در یک عکس رنگی با تعداد بیشتری رنگ سر کار دارید، این پیسکل‌ها می‌توانند دارای مقادیر عددی بزرگتری باشند.

محاسبه تعداد پیکسل

برای محاسبه تعداد پیکسل‌ها در یک عکس می‌توان تعداد پیکسل‌های ستون و سطر را در هم ضرب کرد.

تعداد پیکسل در یک عکس = تعداد پیکسل ستون * تعداد پیکسل سطر

تعداد بیت‌ها در هر پیکسل را با BPP به معنای Bits Per Pixel نشان می‌دهند. در یک عکس سیاه و سفید مقدار بیت‌ها در هر پیکسل می‌تواند صفر و یا یک باشد. اگر بخواهید رابطه‌ای برای آن درنظر بگیرید، می‌توان آن را به‌صورت زیر نوشت.

$$\text{تعداد پیکسل در هر بیت} = (2)^{BPP}$$

تعداد رنگ‌ها

تعداد رنگ‌های مختلف در یک تصویر به عمق رنگ یا بیت در هر پیکسل بستگی دارد. جدول ۱-۱ بیت‌های مختلف در هر پیکسل و مقدار رنگ‌ها را نشان می‌دهد.

جدول ۱-۱

تعداد بیت در پیکسل	تعداد رنگ‌ها
1 BPP	2
2 BPP	4
3 BPP	8
4 BPP	16
5 BPP	32
6 BPP	64
7 BPP	128
8 BPP	256
10 BPP	1024
16 BPP	65536
24 BPP	16777216.7
32 BPP	42949672964294

سایه‌ها

در جدول فوق (۱-۱) می‌توان رشد سریع رنگ‌ها را با افزایش تعداد بیت‌ها در هر پیکسل را دید. یک عکس ۴ بیتی دارای تعداد ۱۶ رنگ است. اگر این عکس خاکستری باشد. دارای ۱۶ سایه رنگ با درجات مختلف خاکستری خواهد بود.

مقادیر رنگ

همانطور که در مورد عکس ۱ بیتی سیاه و سفید گفته شد، مقدار پیکسل برای رنگ سیاه صفر و برای رنگ سفید یک است. مقدار رنگ سفید از رابطه $(2)^{bpp} - 1$ به‌دست می‌آید. برای یک عکس ۴ بیتی رنگ سفید برابر یا ۱۵ است یا برای یک عکس ۸ بیتی رنگ سفید برابر یا ۲۵۵ خواهد شد.

رنگ خاکستری

منظور از رنگ خاکستری مقادیر عددی بین صفر تا عدد انتهای رنگ است. برای مثال، در یک عکس ۸ بیتی که صفر رنگ سیاه و ۲۵۵ رنگ سفید است. رنگ خاکستری محدودهٔ بین این دو رنگ است.

حجم تصویر

برای محاسبه حجم یک تصویر می‌توان با ضرب تعداد پیکسل‌های آن در مقدار بیت در هر یک از پیکسل‌ها، آن را به‌دست آورد.

حجم تصویر = تعداد پیکسل سطر * تعداد پیکسل ستون * تعداد بیت در هر پیکسل

برای مثال، اگر تعداد پیکسل‌های سطر و ستون یک تصویر ۱۶ بیتی به ترتیب ۳۰۰ و ۴۰۰ باشد. اندازه حجم آن برابر ۱۹۲۰۰۰۰ بیت خواهد بود. با تقسیم اندازه حجم بر ۸ می‌توان اندازه را برحسب KB به‌دست آورد.

مفهوم بعد در یک تصویر

بعد در معنا به یک پارامتر یا اندازه گفته می‌شود که برای تعریف ویژگی‌های یک جسم به آن نیاز دارید. وقتی به یک موضوع و یا یک منظره نگاه می‌کنید، چشمانتان تمام جزئیات آن مانند طول، عرض و ارتفاع را با دقت می‌بینند. حال اگر از همان موضوع یا منظره با دوربین یک عکس تهیه کنید. تصویر با آنچه شما می‌بینند تفاوت دارد. دوربین تصویر را به‌صورت سطح ۲ بعدی ولی چشم ما آن را به‌صورت چندبعدی می‌بینند.

برداشت اصولی تصویر

برداشت اصولی تصویر یعنی برداشت و دریافت تصاویر به‌طور مطلوب. عوامل زیادی وجود دارد که می‌تواند در کیفیت عکس نقش مهمی را ایفا کند؛ مانند کادربندی، ترکیب‌بندی، ترکیب رنگ و اشیاء، استفاده از شاتر و دیافراگم و... .

در هنگام دریافت تصاویر، آنها می‌توانند دچار مشکل شوند. گاهی اوقات به‌دلیل سخت‌افزاری، گاهی به دلیل شرایط محیطی، نور و دیگر عوامل می‌توانند دچار افت کیفیت و نویز شوند.

نحوهٔ ورود تصویر به رایانه

برای ورود تصاویر به رایانه می‌توان از روش‌های مختلفی استفاده کرد.

- ✓ اسکنر
- ✓ دوربین‌های دیجیتالی
- ✓ اینترنت
- ✓ Windows

فصل دوم

پردازش تصویر

جعبه ابزار پردازش تصویر

در MATLAB جعبه ابزار پردازش تصویر دارای مجموعه‌ای جامع از الگوریتم‌های استاندارد، توابع و برنامه‌های کاربردی برای پردازش تصویر، آنالیز، تجسم و توسعه الگوریتم است و می‌تواند بهبود عکس، رفع تاری و اِشکال در عکس، خصوصیت تشخیص، کاهش نویز، ثبت تصویر، تبدیل هندسی و... را انجام دهد. علاوه‌بر این، جعبه ابزار پردازش دارای مجموعه‌ای متنوع از انواع عکس شامل HDR، وضوح تصویر، توابع تجسم، امکان جستجو در تصویر، بررسی نواحی پیکسل‌ها، تنظیم کنتراست، ایجاد خطوط کانتور، هیستوگرام و دست‌کاری نواحی دلخواه ROI را دارد. با الگوریتم‌های این جعبه ابزار می‌توان تصاویر تخریب‌شده را بازیابی کرد و ویژگی‌های آن را شناسایی و اندازه‌گیری کرد.

عکس در MATLAB

ساختار اصلی داده در MATLAB آرایه است. آرایه یک مجموعه‌ای مرتب‌شده از عناصر حقیقی یا مختلط است. این مجموعه به‌طور طبیعی برای ارائه عکس‌ها و مجموعه رنگ‌ها و شدت‌ها مناسب است. اکثر تصاویر و عکس‌ها در MATLAB به‌صورت ماتریس ۲ بعدی ذخیره می‌شوند که هر عنصر آن مربوط به یک پیکسل است. برای مثال، عکس متشکل از ۲۰۰ پیکسل در سطر و ۳۰۰ پیکسل در ستون به‌صورت ماتریس ۳۰۰ * ۲۰۰ ذخیره می‌شود.

درک فضاهای رنگی و تبدیل فضای رنگی

MATLAB معمولاً رنگ‌ها را به‌عنوان مقادیر عددی RGB نمایش می‌دهد. علاوه‌بر مدل RGB، مدل‌های رنگی دیگری نیز برای نمایش عددی رنگ‌ها وجود دارد. دلیل وجود فضاهای رنگی مختلف این است که اطلاعات رنگی را به روشی ارائه دهند که برخی از محاسبات راحت‌تر انجام شود و یا به این دلیل که برای شناسایی رنگ‌ها ارائه بصری بهتری دارند. فضای RGB، یک رنگ را به‌عنوان درصدی از رنگ‌های قرمز، سبز و آبی نمایش می‌دهد و یا فضای HSV رنگ را براساس میزان خلوص رنگ، اشباع آن و درخشندگی (شدت یا روشنایی) توصیف می‌کند.

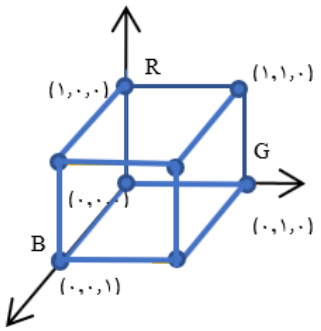
انواع مدل‌های رنگی

انواع مختلفی از فضاها و مدل‌های رنگی در پردازش تصویر استفاده می‌شود. این مدل‌ها می‌تواند شامل RGB, CYMK, Y'UV, YC_bC_r, HSV باشد.

RGB

این مدل از پرکاربردترین مدل‌ها و فضاهای رنگی است. این مدل از سه رنگ نوری اصلی قرمز، سبز و آبی تشکیل شده است. آنچه این مدل رنگی بیان می‌کند این است که هر تصویر رنگی می‌تواند از ترکیب سه رنگ قرمز، سبز و آبی تشکیل شود.

مدل رنگی RGB مبنی‌بر سیستم مختصاتی کارتزین (دکارتی) است و می‌توان آن را بر روی محورهای مختصات ۳ بعدی به‌صورت شکل ۲-۱ ترسیم کرد. برای نمایش این مدل رنگی از فرم (R,G,B) استفاده می‌شود. در این فرم به جای رنگ‌های قرمز، سبز و آبی به ترتیب از اعداد صفر و یک به‌عنوان عدم وجود و وجود رنگ استفاده می‌شود. برای مثال (1,0,0) نشان‌دهندهٔ این است که رنگ قرمز وجود دارد ولی رنگ‌های سبز و آبی وجود ندارد.



شکل ۲-۱

از ترکیب سه رنگ اصلی قرمز، سبز و آبی رنگ سفید حاصل می‌شود. برای نمایش آن از فرم (1,1,1) استفاده می‌شود. از ترکیب دو به دو رنگ‌های اصلی می‌توان رنگ‌های فرعی (مکمل) را به وجود آورد. این مدل رنگی را افزایشی هم می‌نامند. جدول ۲-۱ ترکیب رنگ‌ها را نشان می‌دهد.

جدول ۲-۱

ترکیب رنگ	رنگ حاصل	علائم رنگی
قرمز + آبی	ارغوانی (سایان)	C
آبی + سبز	فیروزه‌ای (ماژنتا)	M
قرمز + سبز	زرد	Y
قرمز + سبز+ آبی	سفید	W
ارغوانی+ سبز	سفید	W

فیروزه‌ای + قرمز	سفید	W

کاربرد مدل رنگی RGB

- ✓ در لامپ‌های تصویر CRT
- ✓ در نمایشگرهای LCD, LED, PLASMA

CMY

این مدل رنگی از سه رنگ فرعی Cyan, Magenta, Yellow استفاده می‌کند. برای به‌دست‌آوردن رنگ‌های فرعی CMY از رنگ‌های اصلی از رابطه زیر استفاده می‌شود.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

در ماتریس فوق اعداد یک به‌منزله رنگ سفید هستند که رنگ‌های اصلی قرمز، سبز و آبی از آن کم می‌شوند تا رنگ‌های CMY به‌دست آیند. به‌عنوان مثال، اگر که یک سطح به رنگ آبی باشد، با تابش رنگ سفید به آن سطح، نوری آبی از آن منعکس می‌شود و رنگ زرد از سفید کم می‌شود (B = 1 – Y).

به این مدل رنگی، کاهشی نیز گفته می‌شود. کاربرد این مدل رنگی در چاپگرها و پرینترها است. رنگ مشکی مورد نیاز از ترکیب سه رنگ مکمل به‌دست می‌آید. این مدل رنگی را CMYK نیز می‌نامند.

Y'UV

این یک مدل و فضای رنگی است که Y′ لومینانس و دو مؤلفه دیگر کرومینانس هستند. از این مدل رنگی در استانداردهای تصاویر رنگی تلویزیون استفاده می‌شود و در سیستم آنالوگ کاربرد دارد.

NTSC – PAL -SCAM

YIQ

این مدل رنگی مخصوص سیستم تلویزیونی ایالات متحده آمریکا است. یکی از مهمترین مزیت آن این است که اطلاعات خاکستری را از داده‌های رنگی جدا می‌کند و می‌توان از همان سیگنال برای هر دو مجموعه رنگی و خاکستری استفاده کرد. در این فضای رنگی، داده عکس شامل Y درخشندگی، I خلوص رنگ و Q اشباع می‌باشد. مؤلفه Y، اطلاعاتی از درجات خاکستری و دو مؤلفه دیگر اطلاعات رنگی (کرومینانس) را تشکیل می‌دهند.

YC_bC_r

در این مدل و فضای رنگی Y′ لومینانس و دو مؤلفه دیگر کرومینانس هستند. این یک مدل رنگی مطلق نیست و عمدتاً در سیستم دیجیتال کاربرد دارد.

HSV

این مدل رنگی برخلاف مدل‌ها و فضای رنگی RGB, CMYK که از رنگ‌های اصلی و مکمل استفاده می‌کنند، به نحوهٔ درک انسان از رنگ‌ها نزدیکتر است و از سه مؤلفه Hue خلوص رنگ، Saturation اشباع و Value مقدار رنگ استفاده می‌کند. این مدل معادل HSI, HSB است. در این دو B روشنایی و I شدت رنگ است.

انواع داده‌های عددی در MATLAB

قبل از شروع به بحث در مورد انواع تصاویر در MATLAB، بهتر است مروری از انواع داده‌های عددی شود.

Numeric Data

این نوع داده عددی در MATLAB شامل اعداد صحیح علامت‌دار، بدون علامت، Single و Double است. این تنوع داده‌ها از نظر میزان فضای ذخیره‌سازی و دامنه عددی با هم متفاوت هستند.

MATLAB به‌طور پیش‌فرض داده عددی را در کلاس double (دقت مضاعف) نگهداری می‌کند.

Integer

جدول ۲-۲ انواع داده‌های عددی صحیح، بازه و میزان فضای اشغال‌شده توسط هریک از آنها را نشان می‌دهد.

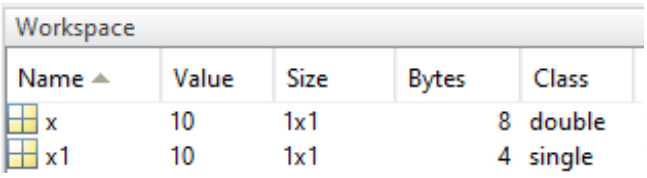
جدول ۲-۲

نام داده	بازه عددی داده	میزان اشغال حافظه (بایت)
int8	-2^7 تا 2^7-1	۱
int16	-2^{15} تا $2^{15}-1$	۲
int32	-2^{31} تا $2^{31}-1$	۴
int64	-2^{63} تا $2^{63}-1$	۸
uint8	0 تا 2^8-1	۱
uint16	0 تا $2^{16}-1$	۲
uint32	0 تا $2^{32}-1$	۴
uint64	0 تا $2^{64}-1$	۸

Floating-point

MATLAB اعداد اعشاری را در قالب double و single نمایش می‌دهد (شکل ۲-۳).

```
>> x=10
x =
    10
>> x1=single(x)
x1 =
  single
    10
```



شکل ۲-۳

به علت تفاوت در Class دو متغیر x, x1، میزان فضای اشغال‌شده توسط هریک از آنها متفاوت است. در جدول ۲-۳ داده‌های اعشاری، بازه و میزان فضای اشغال‌شده را نشان می‌دهد.

جدول ۲-۳

نام داده	بازه عددی داده	میزان اشغال حافظه
Single	1.1755e-38تا 3.4028e+38	۴
double	.2251e-308۲ تا 1.7977e+308	۸

Logical

داده منطقی با استفاده از صفر و یک ارائه می‌شود. بعضی از عملگرها و توابع MATLAB پاسخی منطقی به شرایط می‌دهند که می‌تواند درست یا غلط باشد. برای مثال، یک بودن نتیجه به‌منزلهٔ درست بودن و صفرشدن نتیجه به‌منزله اشتباه شرایط است.

```
>> 4*20 > 6*12
ans =
  logical
   1
```

انواع تصاویر رقومی

تبدیل تصاویر آنالوگ به فرم و شکل عددی و قابل استفاده در رایانه را تصاویر رقومی می‌نامند.

تصاویر در MATLAB به ۴ گروه تقسیم می‌شوند.

۱- Binary
۲- Indexed
۳- GrayScale
۴- TrueColor

Binary

تصاویر باینری، تصاویری هستند که پیکسل‌های آنها می‌توانند فقط دارای دو مقدار صفر و یک باشند و به‌صورت سیاه و سفید نمایش داده شوند. داده‌های این تصاویر به‌صورت آرایه منطقی (Logical) ذخیره می‌شود. برای مثال، عکسی که در ویندوز با فرمت Monochrome ذخیره می‌شود، یک عکس باینری است. برای تهیه این نوع عکس می‌توان از دکمه Print Screen صفحه کیبورد استفاده کرد. با زدن این دکمه یک عکس از صفحه Desktop به حافظه موقت ویندوز می‌رود. با انتخاب یک ویرایشگر عکس (مانند paint) و بازکردن آن و سپس الصاق عکس به آن، عکس به ویرایشگر منتقل می‌شود. از سربرگ File گزینه save یا save as را انتخاب کنید تا پنجره save as باز شود. از منوی کشویی Save as type فرمت Monochrome Bitmap را انتخاب کنید و گزینه save را کلیک کنید تا عکس در قالب باینری ذخیره شود.

Indexed

این نوع تصاویر، اندیس‌گذاری شده (شاخص دار) و به‌عنوان شبه‌رنگی شناخته می‌شوند. در این نوع عکس از یک آرایه به نام x برای داده‌های عکس و یک ماتریس map به‌عنوان نقشه رنگ استفاده می‌شود. کلاس آرایه‌های این تصاویر می‌تواند uint8, uint16, single, double باشد، map به‌صورت یک ماتریس ۳* m و مقادیر آن می‌تواند دارای کلاس Double و بین [0, 1] باشد. مقادیر هر عنصر از آرایه عکس در حقیقت اشاره‌ای به درون ماتریس رنگ دارد تا در حالت رنگی و یا خاکستری پیکسل مربوطه را تعیین کند. یک تصویر Indexed بسته به مقادیر map ممکن است رنگی یا خاکستری باشد. هر سطر ماتریس map بیانگر ترکیبی خاص از رنگ‌های قرمز، سبز و آبی است.

همراه هر تصویر Indexed، ماتریس رنگ مربوطه نیز ذخیره می‌شود. اگر شما تصویری را فراخوانی کنید، ماتریس رنگ آن نیز به‌طور خودکار فراخوانده خواهد شد.

بین مقادیر عناصر آرایه x و تعداد سطرهای ماتریس رنگ map همواره رابطه‌ای وجود دارد و این به نوع کلاس آرایه بستگی دارد. اگر کلاس آرایه از نوع single, double باشد، آرایه x شامل اعدادی بین ۱ تا P است. P طول ماتریس map است. مقدار عددی یک در آرایه x به اولین سطر map و مقدار دو به دومین سطر و مقدار P به P امین سطر اشاره دارد. اگر نوع تصویر یکی از انواع logical, uint8,uint16 باشد، مقدار صفر به اولین سطر، مقدار یک به دومین سطر و مقدار p به p+1 سطر اشاره دارد.

Grayscale

این نوع به‌عنوان تصاویر سطوح خاکستری هستند. این تصاویر می‌توانند دارای آرایه‌های کلاس uint8, uint16, int16, single, double باشند، به‌طوری که مقادیر پیکسل‌ها، مقادیر شدت سطوح را تعیین می‌کنند. یک تصویر خاکستری، یک ماتریس دوبعدی است که عناصر آن معرف شدت روشنایی پیکسل مربوطه در محدوده مشخص می‌باشد. MATLAB برای نمایش تصاویر سطوح خاکستری نیز از map (مشابه حالت اندیس شده) استفاده می‌کند، اما برای ذخیره آنها از map استفاده نمی‌کند. اگر ماتریس از نوع single, double باشد، مقدار صفر برای

رنگ سیاه و مقدار یک برای رنگ سفید است. برای تصاویر از نوع uint8, uint16, int16 مقدار intmin(Class(I)) متناظر است با رنگ سیاه و نوع مقدار intmax(Class(I)) متناظر است با رنگ سفید است.

Truecolor

تصاویر TrueColor یا RGB تصاویری هستند که هر پیکسل آنها با سه عدد مشخص می‌شود. هریک از سه عدد مربوط به‌شدت یکی از رنگ‌های قرمز سبز و آبی است. در MATLAB هر تصویر رنگی به‌صورت آرایه‌های سه‌بعدی m*n*3 ذخیره می‌شود. دارای ۳ ماتریس رنگ می‌باشد که هر ماتریس رنگ مشخص‌کننده شدت یکی از رنگ‌ها برای تمام پیکسل‌های مختلف در عکس است. در این نوع عکس خبری از map نیست. هر تصویر رنگی می‌تواند یکی از انواع داده‌های uint8, uint16, single, double باشد. در حالت single, double هر مؤلفه‌ای رنگ، عددی بین صفر تا یک است. صفر به‌صورت (0, 0,0) و سفید (1, 1,1) نشان داده می‌شود.

به‌طور خلاصه می‌توان مطالب فوق را در قالب جدول ۲-۴ ارائه داد.

جدول ۲-۴

نوع عکس	توضیحات
Binary	**داده این عکس در قالب یک ماتریس m*n منطقی ذخیره می‌شود. مقادیر آن می‌تواند صفر و یا یک باشد. (سیاه و سفید)**
Indexed	**داده این نوع عکس در قالب یک ماتریس m*n عددی ذخیره می‌شود که رنگ عناصر آن در یک Colormap به‌طور مستقیم نشان داده می‌شوند. هر سطر ماتریس map اجزای قرمز، سبز و آبی یک رنگ را مشخص می‌کند.** • **برای آرایه با کلاس single, double مقادیر اعداد صحیح map بین ۱ تا m است.** • **برای آرایه با کلاس logica, uint8, uint16 مقادیر map بین ۱ تا m-1 است.** **Colormap یک ماتریس m*3 از نوع double است.**
Garyscale یا Intendity	**داده این عکس به‌عنوان یک آرایه عددی m*n ذخیره می‌شود که عناصر آن مقادیر شدت را مشخص می‌کند.** • **برای آرایه با کلاس single, double مقادیر بین [0,1] است.** • **برای آرایه با کلاس uint8 مقادیر بین [0,255] است.** • **برای آرایه با کلاس uint16 مقادیر بین [0,65535] است.**

• برای آرایه با کلاس int16، مقادیر بین [32768,32767-] است.	
داده این نوع تصاویر به‌صورت یک آرایه سه‌بعدی m*n*3 ذخیره می‌شود که عناصر آن مقادیر شدت یکی از سه کانال رنگ را مشخص می‌کند. برای تصاویر RGB سه کانال نشانگر سیگنال‌های قرمز، سبز و آبی است. • برای آرایه کلاس single, double مقادیر RGB بین [0,1] است. • برای آرایه با کلاس uint8 مقادیر RGB بین [0,255] است. • برای آرایه با کلاس uint16 مقادیر RGB بین [0,65535] است.	RGB True color

اصول فراخوانی تصاویر در نرم‌افزار MATLAB

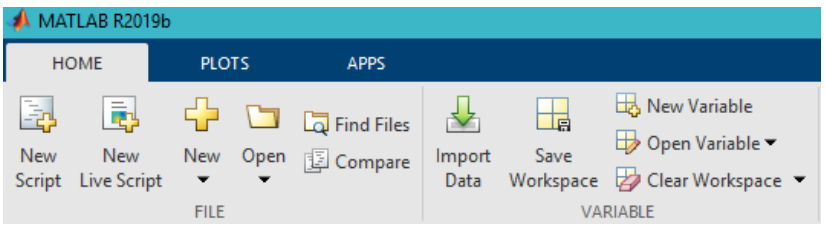
برای فراخوانی یک تصویر به MATLAB از روش‌های زیر استفاده می‌شود.

۱- Import Wizard
۲- تابع Importdata
۳- تابع Load
۴- تابع imread

Import Wizard

ساده‌ترین راه برای واردکردن داده به MATLAB، استفاده از Import Wizard است. برای استفاده از این ابزار نیازی به دانستن فرمت داده عکس نیست. می‌توان به‌سادگی فایل داده‌ها را انتخاب کرد و به‌طور خودکار فایل را وارد کرد.

برای دسترسی به این ابزار باید از مسیر HOME → VARIABLE گزینه Import Data انتخاب شود تا پنجره (شکل ۲-۴) باز شود. برای این کار نیز می‌توان با تایپ تابع uiimport در Command window استفاده کرد.



شکل ۲-۴

Importdata

این یک تابع سطح بالا برای فراخوانی داده است. این تابع داده‌ها را در یک قالب یک ماتریس بارگذاری می‌کند. تصاویر هم از این قاعده مستثنی نیستند.

```
>> y=importdata('rgbl.bmp')

  135×198×3 uint8 array
```

بایستی به این نکته توجه شود که عکس مورد نظر در مسیر جاری (Current Folder) باشد.

Load

از این تابع نیز می‌توان برای فراخوانی برخی از عکس‌ها که در مسیر پیش‌فرض نرم‌افزار قرار دارد استفاده کرد.

مثال: فراخوانی عکس اندیس شده.

```
>> load clown
```

imread

در پردازش تصویر، اصلی‌ترین تابع برای فراخوانی عکس به MATLAB تابع imread است. این تابع داده‌های یک فایل گرافیکی را در قالب یک آرایه به Workspace می‌برد. با این تابع می‌توان اکثر فرمت‌ها را به نرم‌افزار فراخوانی کرد. مانند: TIFF, GIF, JPEG, PNG,... تابع imread را می‌توان در فرم‌ها و قالب‌های مختلفی به‌کار برد.

A=imread(filename) این فرم از تابع، تصویر را به‌وسیله نام فایل می‌خواند. تابع فرمت فایل را از روی محتوای آن به‌دست می‌آورد. اگر فایل عکس، یک فایل چند فریمی باشد، فقط اولین فریم آن را می‌خواند.

A=imread(filename.fmt) این حالت، فایل تصویر را با استفاده از پسوند آن می‌خواند.

[X,map]=imread(___) این فرم از تابع، عکس اندیس‌شده را در X و نقشه رنگ آن را در map قرار می‌دهد.

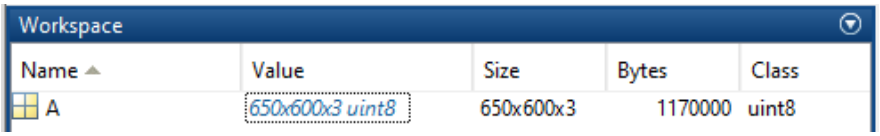
بعد از فراخوانی تصویر داده‌های آن به‌صورت یک آرایه در Workspace قرار می‌گیرد. کلاس تصویر در Workspace بستگی به فرمت و عمق بیت داده آن دارد.

- ✓ اگر عکس در مقیاس خاکستری باشد، آرایه آن به‌صورت m*n است.
- ✓ اگر عکس به‌صورت اندیس‌شده باشد آرایه آن m*n و مقادیر عددی آرایه متناظر با رنگ در map است.
- ✓ اگر عکس رنگی باشد، آرایه آن به‌صورت m*n*3 خواهد بود.
- ✓ اگر عکس چند فریمی باشد. آرایه آن به‌صورت m*n*3*p خواهد بود.

مثال، فراخوانی داده عکس با فرمت JPEG و انتقال به Workspace و ذخیره‌شدن آن به‌عنوان آرایه A (شکل ۲-۵).

تابع عکس را در Workspace به‌عنوان آرایه چندبعدی با کلاس uint8 نشان می‌دهد.

```
A = imread('ngc6543a.jpg');
```



شکل ۲-۵

فراخوانی عکس از ویندوز به MATLAB

برای فراخوانی عکس از محیط ویندوز می‌بایست آدرس عکس، نام به همراه فرمت آن در تابع imread وارد شود تا تابع بتواند آن را شناسایی و فراخوانی کند.

مثال: فراخوانی عکس باینری exam.bmp از ویندوز.

```
>> BW=imread('C:\Users\Hosseini\Desktop\exam.bmp');
```

به‌طور معمول در MATLAB عکس‌های باینری را در متغیر BW، اندیس شده در متغیرهایX, map، خاکستری در متغیر I و رنگی در متغیر RGB ذخیره می‌شوند.

جدول ۲-۵ فرمت فایل‌های قابل پشتیبانی، عمق بیت و نوع داده عکس خروجی آرایه داده در imread را نشان می‌دهد.

جدول ۲-۵

نام فرمت	فرمت عکس	توضیحات	
		عمق بیت	کلاس
BMP	Windows Bitmap	1 Bit	Logical
		4 Bit	Uint8
		8 Bit	Uint8
		16 Bit	Uint8
		24 Bit	Uint8
		32 Bit	Uint8
GIF	Graphics Interchange Format	1 Bit	Logical
		2 Bit to 8 Bit	Uint8
HDF	Hierarchical Data Format	8 Bit	Uint8
		24 Bit	Uint8

JPEG JPG	Joint Photographic Experts Group	8 Bit	Uint8
		12 Bit	Uint16
		24 Bit	Uint16
JP2 JPX	JPEG 2000	1 Bit	Logical
		2 to 8 Bit	Uint8 یا int8
		9 to 16Bit	Uint16یا int16
PBM	Portable Bitmap	1 Bit	Logical
PCX	Paintbrush	1 Bit	Logical
		8 Bit	Uint8
		24 Bit	Uint16
PGM	Portable Graymap	8 Bit	Uint8
		16 Bit	Uint16

نام فرمت	فرمت عکس	توضیحات	
		عمق بیت	کلاس
PNG	Portable Network Graphics	1 Bit	Logical
		2 Bit	Uint8
		4 Bit	Uint8
		8 Bit	Uint8
		16 Bit	Uint16
		28 Bit	Uint8
		48 Bit	Uint16
PPM	Portable Pixmap	تا حدود ۱۶ بیت	Uint8
RAS	Sun™ Raster	1 Bit	Logical
		8 Bit	Uint8
		24 Bit	Uint8
		32 Bit	Uint8
TIFF TIF	Tagged Image File Format	1Bit	Logical
		16 Bit	Uint16

XWD	X Window Dump	1 Bit	Logical
		8 Bit	Uint8
CUR	Cursor File	1 Bit	Logical
		4 Bit	Uint8
		8 Bit	Uint8
ICO	Icon File	1 Bit	Logical
		4 Bit	Uint8
		8 Bit	Uint8

ارسال و بازنویسی عکس

برای ارسال و بازنویسی یک عکس از تابع imwrite استفاده می‌شود. در هنگام استفاده از این تابع، MATLAB به‌طور پیش‌فرض عمل ذخیره‌سازی را انجام می‌دهد و به‌طور خودکار عمق بیت را انتخاب و کلاس را به uint8 تبدیل می‌کند. اکثر فایل‌های استفاده شده در MATLAB ۸ بیتی هستند و نیاز به داده با دقت مضاعف ندارند. یک قانون استثناء برای ذخیره‌کردن داده‌ها با uint8 وجود دارد که عکس‌های با فرمت PNG, TIFF ممکن است با uint16 ذخیره شوند. در حالی‌که این دو فرمت داده‌های ۱۶ بیتی را هم پشتیبانی می‌کنند. ممکن است MATLAB به‌طور خودکار عملیات را انجام دهد و نوع داده را uint16 درنظر بگیرد. تابع imwrite شبیه به imread دارای فرم‌های مختلفی است.

عملکرد تابع با عکس‌های خاکستری، رنگی و منطقی

imwrite(A,filename): این تابع داده عکس A را با نام تعیین‌شده در فرمتی که خود براساس محتویات آن تشخیص می‌دهد، در مسیر جاری بازنویسی می‌کند و ذخیره می‌کند. عمق بیت عکس خروجی بستگی به نوع داده عکس A و فرمت آن دارد.

عملکرد تابع imwrite برای اکثر فرمت‌ها به‌صورت زیر است.

- ✓ اگر عکس A از نوع uint8 باشد، خروجی تابع ۸ بیتی است.
- ✓ اگر عکس A از نوع uint16 باشد و فرمت فایل خروجی ۱۶ بیت (JPEG, PNG, TIFF) را پشتیبانی کند، خروجی تابع ۱۶ بیت است. اگر فرمت فایل خروجی ۱۶ بیت را پشتیبانی نکند، تابع خطایی را برمی‌گرداند.
- ✓ اگر عکس A، خاکستری یا RGB با انواع داده single, double باشد، تابع فرض می‌کند که محدوده دینامیکی [0,1] است و به‌طور خودکار داده را ۲۵۵ مقیاس‌دهی می‌کند و به‌عنوان فایل ۸ بیتی ارسال می‌کند. اگر داده A، single باشد، A را قبل از نوشتن به GIF یا TIFF به double تبدیل می‌کند.
- ✓ اگر عکس A از نوع منطقی باشد، imwrite فرض می‌کند که داده باینری است و اگر فرمت آن اجازه دهد، آن را با عمق بیت ۱ بازنویسی می‌کند.

عملکرد تابع با عکس‌های indexed

در صورتی‌که عکس از نوع اندیس شده باشد، باید map را به‌عنوان آرگومان ورودی اضافه کرد.

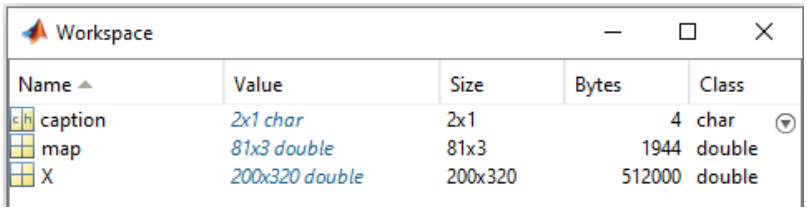
imwrite(X,map,filename): این تابع داده عکس اندیس شده X و map (نقشه رنگ) وابسته به آن را در filename بازنویسی و ذخیره می‌کند.

- ✓ اگر X دارای داده double یاsingle باشد، imwrite شاخص‌ها را به‌وسیله کم‌کردن عدد یک از هر عنصر و سپس داده را به‌عنوان uint8 بازنویسی می‌کند. اگر داده عکس X، single باشد، آن را قبل از بازنویسی به GIF, TIFF تبدیل می‌کند.
- ✓ می‌توان تابع را با پارامترهای اضافی برای فرمت‌های خروجی PBM, JPEG, HDF, GIF, PPM, PNG با استفاده از یک یا چند آرگومان جفت مقدار تعیین و بازنویسی کرد.

مثال: عکس اندیس‌شده clown را فراخوانی و با نام photo و پسوند PNG ذخیره شود.

با فراخوانی عکس clown، آرایه عکس در متغیر X و نقشه رنگ آن (map) در Workspace قرار می‌گیرد (شکل ۲-۶).

```
>> load clown
```



شکل ۲-۶

```
>> imwrite(X,map,'photo.png')
```

بازنویسی عکس اندیس‌شده با پسوند PNG. به‌طور پیش‌فرض عکس در مسیر جاری قرار می‌گیرد.

مثال: تغییر در نقشه رنگ (map) عکس اندیس مثال بالا.

برای تغییر در رنگ تصاویر اندیس‌شده می‌توان یک نقشه رنگ (map) جدید ساخت و آن را به همراه آرایه اصلی عکس استفاده کرد. map یک ماتریس m*3 با داده‌هایی از نوع double است. map، تصویر clown یک ماتریس 81*3 است. با استفاده از تابع rand می‌توان یک ماتریس به ابعاد 100*3 با داده‌های تصادفی بین صفر و یک ساخت.

```
>> map1=rand(100,3);
```

حال با بازنویسی مجدد تصویر با map1، خروجی آن از نظر نقشه رنگی تغییر می‌کند.

```
>> imwrite(X,map1,'photo1.png')
```

مثال: تغییر فرمت و عمق بیت در عکس

برخی از فایل‌های گرافیکی را می‌توان با پارامترهای اضافی مثل تعیین عمق بیت و... ذخیره کرد.

```
>> load clown
>> imwrite(X,map,'clown.bmp');
>> imwrite(X,map,'clown.png','BitDepth',8);
```

جدول ۲-۶ فرمت‌های فایل خروجی عکس‌ها و عمق بیت حاصل از تابع imwrite را نشان می‌دهد.

جدول ۲-۶

نام فرمت	فرمت خروجی عکس	توضیحات	
BMP	Windows Bitmap	**1 Bit, 8 Bit, 24 Bit بدون فشرده‌سازی**	
GIF	Graphics Interchange Format	**8 Bit**	
HDF	Hierarchical Data Format	**عکس 8 Bit با یا بدون نقشه رنگ**	
JPEG JPG	Joint Photographic Experts Group	**8 Bit, 12 Bit, 24 Bit**	
نام فرمت	**فرمت خروجی عکس**	**توضیحات**	
JP2 JPX	JPEG 2000	**1Bit, 8 Bit, 16 Bit**	
PBM	Portable Bitmap	**1 Bit**	
PCX	Paintbrush	**8 Bit**	
PGM	Portable Graymap	**1 Bit, 8 Bit, 24 Bit**	
PNG	Portable Network Graphics	1 Bit	**Logical- Grayscale**
		2 Bit	**Uint8 Grayscale**
		4 Bit	**Uint8 Grayscale**
		8 Bit	**Uint8 Grayscale**
		16 Bit	**Uint16 Grayscale**
		28 Bit	**Uint8 RGB**
		48 Bit	**Uint16 RGB**

PNM	Portable Any Map	**هریک از فرمت‌های PGM/ PPM/ PBM به‌طور خودکار انتخاب می‌شود.**	
PPM	Portable Pixmap	**قابلیت پیاده‌سازی تا حدود ۱۶ بیت در هر مؤلفه رنگ**	
RAS	Sun™ Raster	1 Bit	**Bitmap–Logical**
		8 Bit	**Indexed – Uint8**
		24 Bit	**RGB – Uint8**
		32 Bit	**RGB – Uint8**
TIFF TIF	Tagged Image File Format	**1Bit, 8 Bit, 16 Bit, 24 Bit, 48 Bit عکس‌های بدون فشرده‌سازی**	
XWD	X Window Dump	1 Bit	**Logical**
		8 Bit	**Uint8**

Colormap

Colormap یک ماتریس از مقادیر بین ۰ و یک است که رنگ را برای اشیاء گرافیکی (سطح، عکس) تعریف می‌کند. MATLAB اشیاء را به‌وسیلهٔ نگاشت مقادیر داده به رنگ در Colormap می‌کشد. Colormap ها می‌توانند دارای هر طولی باشند، ولی باید دارای سه ستون باشند. هر سطر در ماتریس Colormap یک رنگ را با استفاده از بردار RGB تعریف می‌کند.

برای تعریف یک Colormap (نقشه رنگ) به‌عنوان یک طرح رنگی جدید باید از نام Colormap ها استفاده کرد. نام Colormap یک نقشه رنگ تعریف‌شده به محور جاری اضافه می‌کند. می‌توان یک Colormap سفارشی ساخت و یا اینکه از انواع پیش تعریف‌شده استفاده کرد. برای مثال، تابع colormap(copper(10)) با استفاده از نقشه رنگ copper نقشه رنگ محور جاری را به یک بخش ۱۰ رنگی از copper تنظیم می‌کند.

جدول ۲-۷ لیست colormap پیش تعریف‌شده در MATLAB را نشان می‌دهد.

جدول ۲-۷

ردیف	نام Colormap	ردیف	نام Colormap
۱	parula	۱۰	**gray**
۲	jet	۱۱	**bone**
۳	hsv	۱۲	**copper**

۴	hot	۱۳	**pink**
۵	cool	۱۴	**lines**
۶	spring	۱۵	**colorcube**
۷	summer	۱۶	**prism**
۸	autumn	۱۷	**Flag**
۹	winter	۱۸	**white**

جدول ۲-۸ لیست مقادیر سه‌گانه RGB را برای رنگ‌های مشترک نشان می‌دهد.

جدول ۲-۸

رنگ	بردار RGB
yellow	**[1 1 0]**
magenta	**[1 0 1]**
cyan	**[0 1 1]**
red	**[1 0 0]**
green	**[0 1 0]**
blue	**[0 0 1]**
white	**[1 1 1]**
black	**[0 0 0]**

برای ویرایش یک Colormap می‌توان از تابع colormapeditor استفاده کرد.

نمایش تصاویر گرافیکی در MATLAB

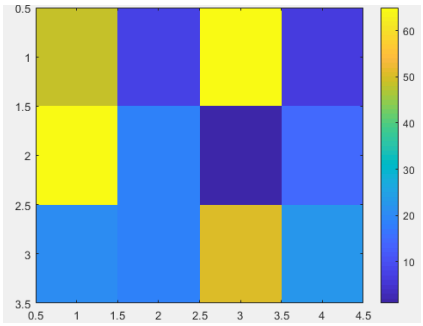
برای نمایش آرایه‌ها و فایل‌های گرافیکی در MATLAB از چندین تابع می‌توان استفاده کرد.

image ()

از این تابع برای نمایش یک آرایه در قالب یک تصویر استفاده می‌شود. هر عنصر آرایه، یک رنگ را برای یک پیکسل از عکس مشخص می‌کند. عکس حاصل یک شبکه m*n از پیکسل‌ها است، به‌طوری که m برابر تعداد سطر و n برابر تعداد ستون آرایه است. شاخص‌های سطر و ستون عناصر پیکسل‌های مربوطه را تعیین می‌کند.

مثال: نمایش ماتریس A (شکل ۲-۷).

```
>> A = [48 7 100 6; 80 18 0 14; 20 18 50 22];
>> image(A)
>> colorbar
```



شکل ۲-۷

imagesc()

این تابع عکس‌ها را با رنگ‌های مقیاس‌دهی شده نمایش می‌دهد، به‌عبارت دیگر داده‌های عکس را برای تمام محدوده از colormap مقیاس‌دهی می‌کند و عکس را نمایش می‌دهد.

مثال: نمایش عکس رنگی (شکل ۲-۸).

```
>> RGB=imread('onion.png');
>> imagesc(RGB)
```

شکل ۲-۸

imshow

در تابع imshow عکس ورودی می‌تواند باینری، اندیس‌شده، خاکستری و رنگی باشد.

در نمایش عکس خاکستری تابع imshow، به‌طور پیش‌فرض محدوده نمایش را برای داده‌های عکس، Figure، محور و خصوصیت اشیاء فراهم می‌کند. در عکس خاکستری در صورتی که از فرم imshow(I, [low, high]) استفاده شود، مقادیر زیر low به رنگ سیاه و بالای high به رنگ سفید نمایش داده می‌شود، ولی اگر از این فرم imshow(I, []) استفاده شود، تابع محدوده مقادیر را در بازه (min(I(:)),max(I(:)) تنظیم می‌کند.

مثال: فراخوانی عکس خاکستری و نمایش آن (شکل ۲-۹).

```
>> I=imread('cameraman.tif');
>> imshow(I)
>> figure, imshow(I,[50 170])
```

شکل ۲-۹

استفاده از فرم imshow(filename)، فایل تصویر را نمایش می‌دهد، ولی هیچ داده‌ای از تصویر را به Workspace انتقال نمی‌دهد. در هنگام استفاده از تابع imshow امکان نمایش تصویر همراه با تغییر در خروجی آن نیز وجود دارد. برای مثال، استفاده از تابع زیر برای بزرگنمایی ۲ برابر تصویر.

```
>> imshow(I,'InitialMagnification',200)
```

تبدیلات تصاویر

در پردازش تصویر تبدیل تصاویر و عکس‌ها به انواع مختلف بسیار مهم و کارآمد است و جعبه ابزار Image Processing چندین تابع را دارد که تبدیلات عکس‌ها را انجام می‌دهد.

تابع dither

با استفاده از این تابع می‌توان عکس رنگی را به اندیس شده و عکس خاکستری را به باینری تبدیل کرد. این تابع سبب افزایش وضوح ظاهری رنگ می‌شود.

۱- X = dither (RGB,map)

این فرم از تابع عکس رنگی را به اندیس‌شده تبدیل می‌کند. این تابع عمل را به‌وسیلهٔ کاهش رنگ‌ها در نقشه رنگ map انجام می‌دهد. باید توجه داشته باشید که map ساخته شود.

مثال: تبدیل عکس رنگی به اندیس شده به‌وسیله map ایجاد شده (شکل ۲-۱۰).

```
>> RGB=imread('onion.png');
>> map=rand(100,3);
>> X=dither(RGB,map);
>> imshow(X,map)
```

شکل ۲-۱۰

۲- WB = dither(I)

این تابع عکس خاکستری را به باینری تبدیل می‌کند (شکل ۲-۱۱).

```
>> I=imread('cameraman.tif');
>> BW=dither(I);
>> imshow(BW)
```

شکل ۲-۱۱

تبدیل عکس رنگی به اندیس شده

برای انجام این تبدیل از تابع rgb2ind به فرم‌های مختلف استفاده می‌شود. این تابع را می‌توان به سه روش استفاده کرد.

۱- کمی‌سازی (کوانتیزاسیون) یکنواخت [x,map] = rgb2ind(RGB,tol)

این قالب از تابع، عکس رنگی را به اندیس شده با استفاده از کمی سازی یکنواخت با تلورانس tol و کاهش رنگ تبدیل می‌کند. مقدار tol بین ۰ تا ۱ است. حداقل تعداد رنگ map از رابطه (floor(1/tol)+1) به‌دست می‌آید.

مثال: تبدیل عکس رنگی به اندیس‌شده (شکل ۲-۱۲).

```
>> RGB=imread('peppers.png');
>> [X,map]=rgb2ind(RGB,0.2);
>> imshow(X,map)
```

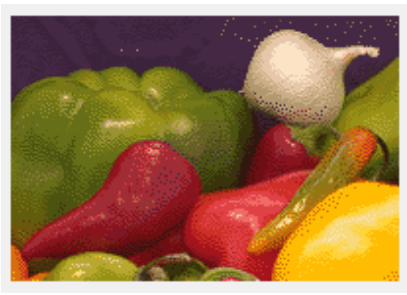
شکل ۲-۱۲

۲- حداقل کمی‌سازی (کوانتیزاسیون) واریانس [x,map] = rgb2ind(RGB,Q)

این فرم از تابع، عکس رنگی را به اندیس شده با اختصاص یک نقشه رنگ map به روش حداقل کوانتیزاسیون واریانس با تعیین تعداد رنگ با Q و کاهش رنگ تبدیل می‌کند. Q عدد مثبت است و می‌تواند کمتر از ۶۵۵۳۵ باشد.

مثال: تبدیل عکس رنگی به اندیس شده (شکل ۲-۱۳).

```
>> RGB=imread('onion.png');
>> [X,map]=rgb2ind(RGB,32);
>> imshow(X,map)
```



شکل ۲-۱۳

۳- نقشه رنگ تقریبی [x,map] = rgb2ind(RGB,inmap)

این حالت از تابع، عکس رنگی را با استفاده از الگوریتم‌های نقشه رنگ معکوس با تعریف نقشه رنگ جدید در متغیر inmap و کاهش رنگ تبدیل می‌کند. inmap یک ماتریس نقشه رنگ m*3 است که می‌بایست آن را ایجاد کرد.

تبدیل عکس اندیس شده به رنگی

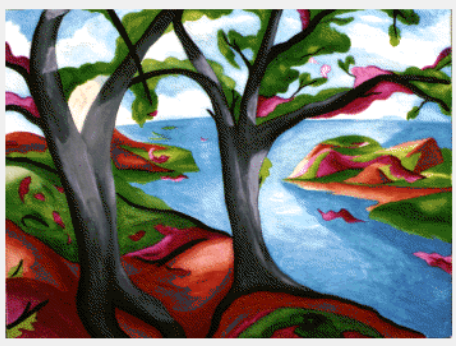
برای تبدیل عکس اندیس شده به رنگی از تابع ind2rgb استفاده می‌شود.

RGB = ind2rgb(X,map)

این تابع عکس اندیس شده و map مربوطه به آن را به رنگی تبدیل می‌کند. عکس ورودی X می‌تواند uint8, uint16, single, double باشد. RGB یک آرایه double است.

مثال: تبدیل عکس اندیس‌شده به رنگی (شکل ۲-۱۴)

```
>> load trees
>> RGB=ind2rgb(X,map);
>> imshow(RGB)
```



شکل ۲-۱۴

تبدیل عکس رنگی یا اندیس‌شده به خاکستری

برای انجام این تبدیلات از تابع rgb2gray استفاده می‌شود.

I= rgb2gray (RGB) -۱

این تابع عکس رنگی را به خاکستری تبدیل می‌کند. تابع این عمل را به وسیلهٔ حذف اطلاعات hue, saturation را تا زمان حفظ لومینانس انجام می‌دهد.

مثال: تبدیل عکس رنگی به خاکستری (شکل ۲-۱۵).

```
>> RGB=imread('peppers.png');
>> I=rgb2gray(RGB);
>> imshow(I)
```

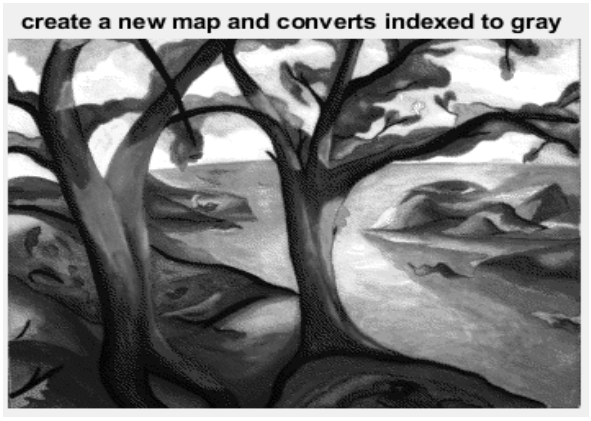
شکل ۲-۱۵

۲- NEWMAP = rgb2gray(map)

این تابع یک map جدید خاکستری معادل map رنگی موجود را تولید می‌کند. در اینجا عملکرد تابع بدین‌صورت است که map رنگی عکس اندیس‌شده را به map خاکستری تبدیل می‌کند.

مثال: تبدیل عکس رنگی اندیس‌شده به خاکستری اندیس‌شده (شکل ۲-۱۶).

```
>> load trees
>> NEWMAP=rgb2gray(map);
>> imshow(X,NEWMAP)
```



شکل ۲-۱۶

اگر عکس ورودی RGB دارای نوع داده uint8, uint16, single, double باشد، خروجی آن عکس دارای همان نوع داده خواهد بود. نوع داده colormap ورودی و خروجی double است.

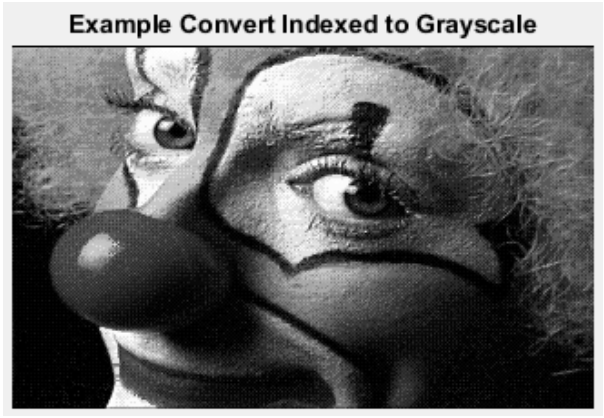
تبدیل عکس اندیس‌شده به خاکستری

I = ind2gray(X,map)

این تابع عکس اندیس‌شده X را با نقشه رنگ موجود map به یک عکس خاکستری تبدیل می‌کند. این تابع اطلاعات hue, saturation عکس اندیس‌شده از بین می‌برد و این عمل را تا زمان حفظ لومینانس تصویر انجام می‌دهد.

مثال: تبدیل عکس اندیس‌شده به خاکستری (شکل ۲-۱۷).

```
>> load clown
>> I=ind2gray(X,map);
>> imshow(I)
```



شکل ۲-۱۷

این تابع، عملکرد تبدیل به خاکستری را با تابع rgb2ntsc انجام می‌دهد و مؤلفه‌های hue, saturation را روی صفر تنظیم می‌کند و نقشه رنگ map را می‌سازد. تابع ind2gray، شاخص X عکس اندیس‌شده را با مقادیر شدت خاکستری در map جایگزین می‌کند.

تبدیل عکس خاکستری و باینری به اندیس‌شده

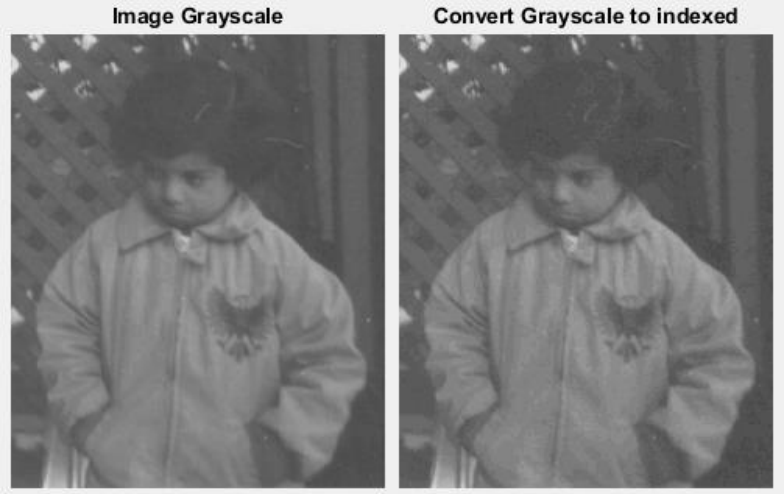
این تبدیلات را می‌توان با استفاده از تابع gary2ind انجام داد.

۱- [X, map] = gray2ind(I,n)

در این فرم n اندازه نقشه رنگ map را مشخص می‌کند. n باید یک عدد صحیح بین ۰ تا ۶۵۵۳۵ باشد. در صورت عدم ورود n، به‌طور پیش‌فرض مقدار آن ۶۴ درنظر گرفته می‌شود.

مثال: تبدیل عکس خاکستری به اندیس‌شده (شکل ۲-۱۸).

```
I=imread('pout.tif');
imshow(I)
[X,map]=gray2ind(I,32);
figure, imshow(X,map)
```



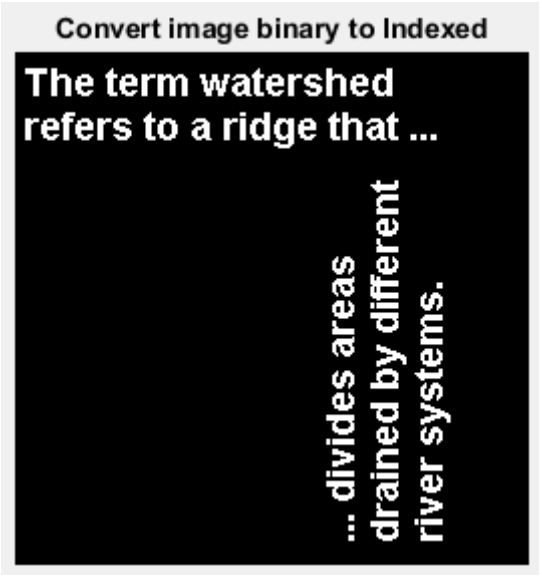
شکل ۲-۱۸

۲- [X, map] = gray2ind(BW,n)

برای تبدیل عکس باینری به اندیس‌شده از این فرم استفاده می‌شود. در عکس باینری مقدار پیش‌فرض n برابر با ۲ است.

مثال: تبدیل عکس باینری به اندیس‌شده (شکل ۲-۱۹).

```
>> BW=imread('text.png');
>> [X,map]=gray2ind(BW,64);
>> imshow(X,map)
```



شکل ۲-۱۹

اندازه عکس اندیس‌شده به‌دست‌آمده با عکس ورودی (باینری و خاکستری) یکی است. اگر طول نقشه رنگ برابر یا کمتر از ۲۵۶ باشد، کلاس عکس خروجی uint8 است در غیر این‌صورت کلاس uint16 خواهد بود.

تابع Grayslice

این تابع عکس خاکستری را با استفاده از تعریف آستانه چند سطحی تقریبی، به عکس اندیس‌شده تبدیل می‌کند. این تابع به‌طور خودکار سطح آستانه را بر پایه n که یک عدد مثبت است تعریف می‌کند.

X = grayslice(I,n)

اگر عدد n را وارد کنید، مقادیر آستانه برای چندین سطح آستانه از روش زیر محاسبه می‌شود.

$$max_{intensity} \times \left\{ \frac{0}{N}, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N} \right\}.$$

$Max_{intensity}$ حداکثر مقدار شدت ممکن بسته به نوع داده عکس خاکستری ورودی است.

- ✓ برای uint8 مقدار $Max_{intensity}$ برابر با ۲۵۵ است.
- ✓ برای uint16, int16 مقدار $Max_{intensity}$ برابر با 65535 است.
- ✓ برای single, double مقدار $Max_{intensity}$ برابر با ۱ است.

مثال: تبدیل عکس خاکستری به اندیس‌شده (شکل ۲-۲۰).

```
>> I=imread('cameraman.tif');
>> imshow(I)
>> X=grayslice(I,32);
>> map=copper(100);
>> figure, imshow(X,map)
```

شکل ۲-۲۰

می‌توان عکس اندیس‌شده را به‌وسیلهٔ آستانه چند سطحی با استفاده از thresholds که یک بردار است به‌دست آورد.

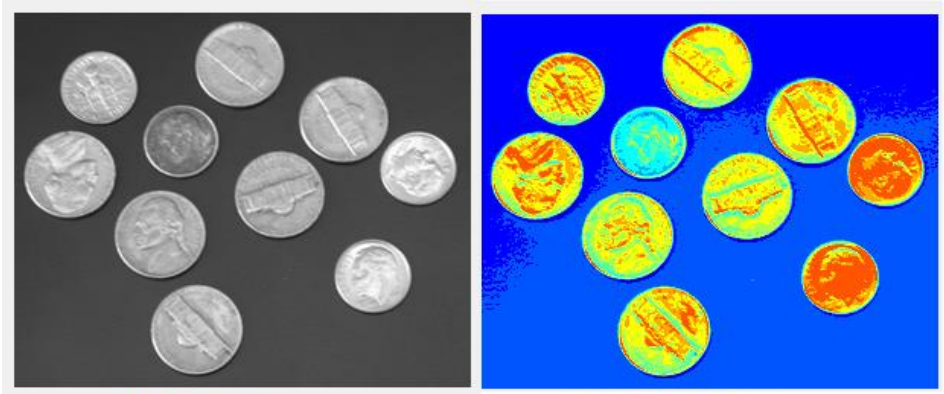
X = grayslice(I,thresholds)

تعداد مقادیر مورد استفاده برای آستانه چند سطحی برابر با طول thresholds است. انتخاب مقادیر آستانه بسته به نوع داده عکس ورودی دارد.

- ✓ مقادیر آستانه برای uint8 می‌تواند بین [0 255] باشد.
- ✓ مقادیر آستانه برای uint16, int16 می‌تواند بین [0 65535] باشد.
- ✓ مقادیر آستانه برای single, double می‌تواند بین [0 1] باشد.

مثال: تبدیل عکس رنگی به اندیس‌شده با استفاده از یک بردار آستانه (شکل ۲-۲۱).

```
>> I = imread('coins.png');
>> imshow(I)
>> thresholds = [45 65 84 108 134 157 174 189 206 228];
>> X = grayslice(I,thresholds);
>> m = double(max(X(:)));
>> figure, imshow(X,colormap(jet(m)))
```



شکل ۲-۲۱

ماتریس عکس اندیس‌شده خروجی هم سایز با عکس ورودی است. کلاس عکس خروجی بسته به تعداد مقادیر آستانه دارد.

- اگر تعداد مقادیر آستانه کمتر از ۲۵۶ باشد، کلاس خروجی uint8 است. در این مورد بازه مقادیر شدت در X بین [0 n] یا [0 length(thresholds)] است.
- اگر تعداد مقادیر آستانه بزرگتر و یا برابر با ۲۵۶ باشد، کلاس خروجی double است. در این مورد بازه مقادیر شدت در X بین [0 n+1] یا [0 length(thresholds)+1] است.

تابع graythresh

T = graythresh(I)

این تابع یک آستانه هدف از عکس خاکستری را با استفاده از روش Otsu's محاسبه می‌کند. این روش یک آستانه‌ای را انتخاب می‌کند که واریانس داخلی آستانه‌های پیکسل‌های سیاه و سفید را به حداقل برساند. آستانه هدف می‌تواند با تابع imbinarize برای تبدیل عکس خاکستری به باینری استفاده شود.

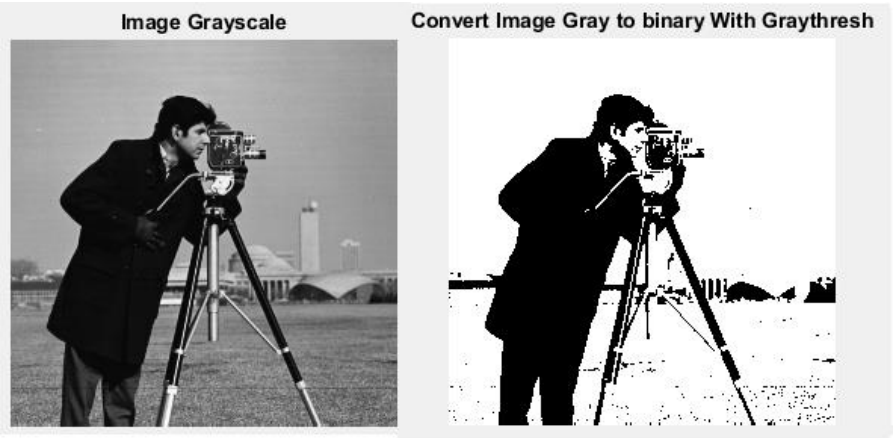
مثال: تبدیل عکس خاکستری به باینری با آستانه مشخص (شکل ۲-۲۲).

```
>> I=imread('cameraman.tif');
>> imshow(I)
>> level=graythresh(I)

level =

    0.3451

>> BW=imbinarize(I,level);
>> figure, imshow(BW)
```



شکل ۲-۲۲

تبدیل سایر عکس‌ها به باینری

تابع im2bw

این تابع می‌تواند عکس‌های رنگی، خاکستری و اندیس‌شده را به باینری تبدیل کند. این تابع تبدیلات را بر پایه مقدار آستانه threshold انجام می‌دهد. سازندگان MATLAB پیشنهاد استفاده از imbinarize را به‌جای im2bw می‌دهند.