دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

گزارش تکلیف اول درس پردازش تصویر رقمی

دانشجو:

سید احمد نقوی نوزاد

ش-د: ۹۴۱۳۱۰۶۰

استاد:

دکتر رحمتی

بهار ۹۵

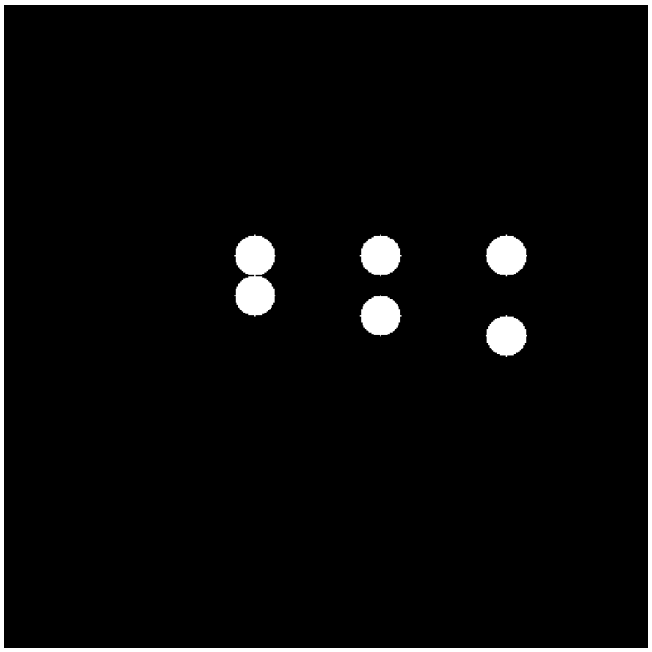
جواب سوال ۱

آشنائی با نحوه‌ی شکل‌گیری عکس (Image Formation)

قسمت الف:

شکل ۱-۱: یک تصویر با ابعاد ۵۱۲*۵۱۲ با مقدار شدت روشنائی صفر برای هر پیکسل

قسمت ب:

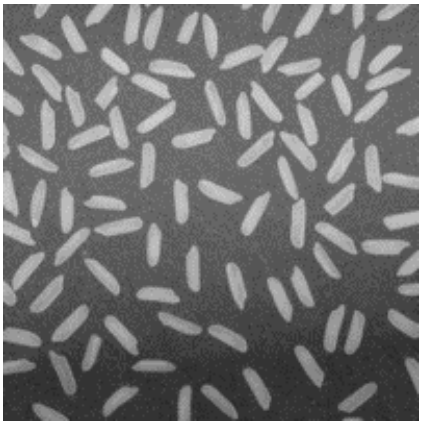


شکل ۲-۱: تصویر قسمت الف با دایره‌های سفید ایجادشده در مراکز (۲۰۰و۲۰۰)، (۲۰۰و۲۳۲)، (۳۰۰و۲۰۰)، (۳۰۰و۲۴۸)، (۴۰۰و۲۰۰) و (۴۰۰و۲۶۴)

جواب سوال ۲

آشنائی با توابع MATLAB جهت افزایش خوانائی و بازدهی پیاده‌سازی‌ها

قسمت الف:

شکل ۱-۲: سمت راست، تصویر cameraman.jpg می‌باشد که با استفاده از تابع imresize متلب به ابعاد تصویر rice.png تغییر اندازه پیدا کرده است.

تابع imresize، آرگومان‌های دریافتی‌اش شامل عکس ورودی جهت تغییر ابعاد و نیز ابعاد جدید در قالب یک بردار می‌باشد. استفاده از تابع size در اینجا از آن جهت مفید است که می‌توان با استفاده از آن ابعاد عکس مقصد را به دست آورد.

قسمت ب:


شکل ۲-۲: سمت راست، تصویر حاصل از جمع نظیر به نظیر پیکسل‌های دو تصویر cameraman.jpg و rice.png با استفاده از عملگر + به دست آمده است و در سمت چپ نیز تصویر حاصل با استفاده از تابع imadd متلب حاصل شده است که همانطور که قابل مشاهده است بین دو تصویر از لحاظ ظاهری هیچ تفاوتی مشاهده نمی‌شود و البته از لحاظ ماتریسی نیز، درایه‌های نظیر به نظیر دو ماتریس کاملا با یکدیگر برابرند و این مسئله با استفاده از دستور زیر چک شده و صحت دارد:

```
>> if imAdd01 == imAdd02; disp('Both images are equal!'); end;
Both images are equal!
```

نکته‌ی قابل توجه در مورد تابع imadd و البته عملگر + این است که در صورت تجاوز حاصل‌جمع از محدوده‌ی قابل نمایش برای شدت روشنائی یک پیکسل، خود نرم‌افزار متلب آن را truncate کرده و مقادیر اعشاری نیز گرد شده و در نهایت نتایج حاصله یکسان خواهند بود.

قسمت ج:

شکل ۳-۲: سمت راست، تصویر حاصل از تفریق نظیر به نظیر پیکسل‌های دو تصویر cameraman.jpg و rice.png با استفاده از عملگر - به دست آمده است و در سمت چپ نیز تصویر حاصل با استفاده از تابع imsubtract متلب حاصل شده است که همانطور که قابل مشاهده است بین دو تصویر از لحاظ ظاهری هیچ تفاوتی مشاهده نمی‌شود و البته از لحاظ ماتریسی نیز، درایه‌های نظیر به نظیر دو ماتریس کاملا با یکدیگر برابرند و این مسئله با استفاده از دستور ذیل بررسی شده و صحت تام دارد:

```
>> if imSubt01 == imSubt02; disp('Both images are equal!'); end;
Both images are equal!
```

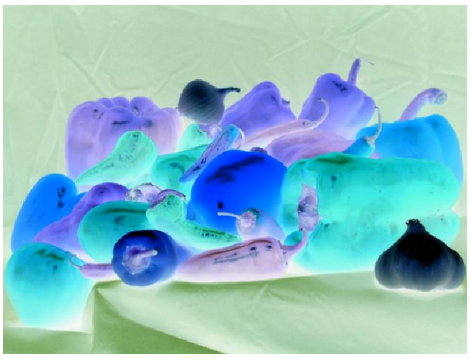
نکته‌ی قابل توجه در مورد تابع imsubtract و البته عملگر - این است که در صورت تجاوز حاصل‌تفریق از محدوده‌ی قابل نمایش برای شدت روشنائی یک پیکسل خود نرم‌افزار متلب آن را truncate کرده و مقادیر اعشاری نیز گرد شده و در نهایت نتایج حاصله یکسان خواهند بود.

قسمت د:

شناسه‌های زیر هم به عنوان نوع داده به کار رفته و هم به عنوان تابع مبدل نوع داده به نوع مربوطه استفاده می‌شوند:

`double`	ورودی را به دقت با ۴ رقم اعشار تبدیل می‌کند
`int8`	ورودی را به یک عدد صحیح علامت‌دار ۸ بیتی تبدیل می‌کند
`int16`	ورودی را به یک عدد صحیح علامت‌دار ۱۶ بیتی تبدیل می‌کند
`int32`	ورودی را به یک عدد صحیح علامت‌دار ۳۲ بیتی تبدیل می‌کند
`int64`	ورودی را به یک عدد صحیح علامت‌دار ۶۴ بیتی تبدیل می‌کند
`uint8`	ورودی را به یک عدد صحیح بی‌علامت ۸ بیتی تبدیل می‌کند
`uint16`	ورودی را به یک عدد صحیح بی‌علامت ۱۶ بیتی تبدیل می‌کند
`uint32`	ورودی را به یک عدد صحیح بی‌علامت ۳۲ بیتی تبدیل می‌کند
`uint64`	ورودی را به یک عدد صحیح بی‌علامت ۶۴ بیتی تبدیل می‌کند
`cast`	کلاس متغیر ورودی را به کلاس متفاوتی تبدیل می‌کند

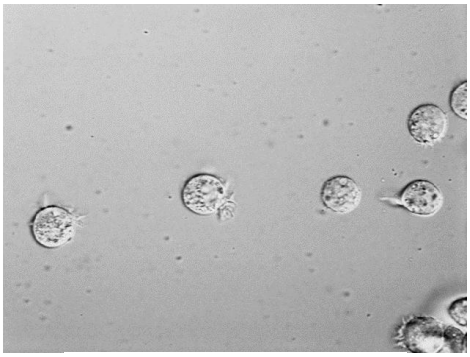
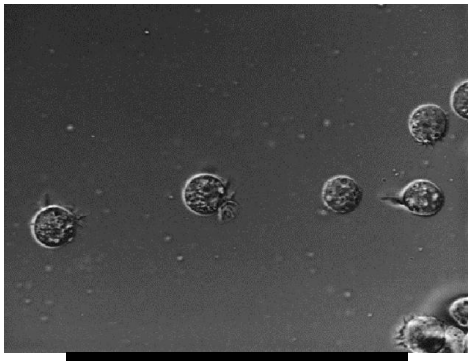
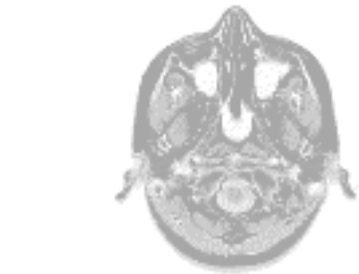
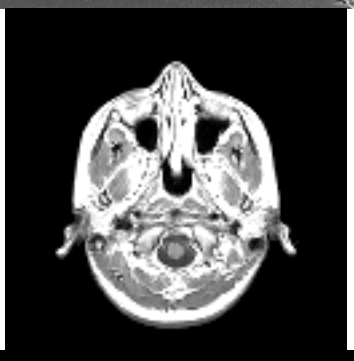
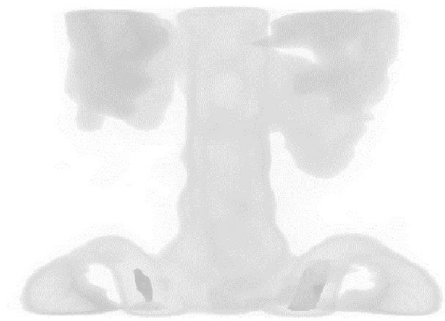
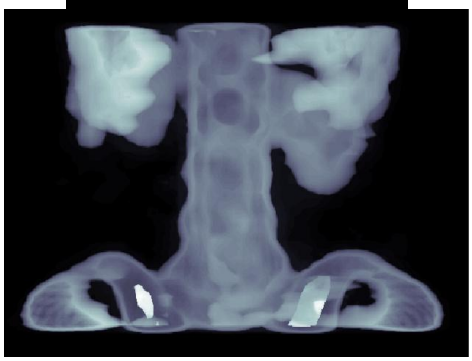
قسمت ه:




شکل ۴-۲: سمت راست، تصویر حاصل با استفاده از تابع imcomplement متلب حاصل شده است که در آن مقدار هر پیکسل از مقدار بیشینه‌ی پشتیبانی‌شده توسط کلاس مربوط به تصویر، کم شده و به عنوان مقدار خروجی استفاده شده است و به عبارتی در تصویر نهائی, نواحی تیره روشن‌تر شده و نواحی روشن هم تیره‌تر می‌شوند.

همانطور که قابل مشاهده است تصویر نهائی شباهت زیادی با عکس‌های نگاتیو دوربین‌های عکس‌برداری سنتی دارد که البته با انجام عملیات شیمیائی خاصی در تاریکخانه‌ی مخصوص آن‌ها را به تصویر نهائی مبدل می‌نموده‌اند.

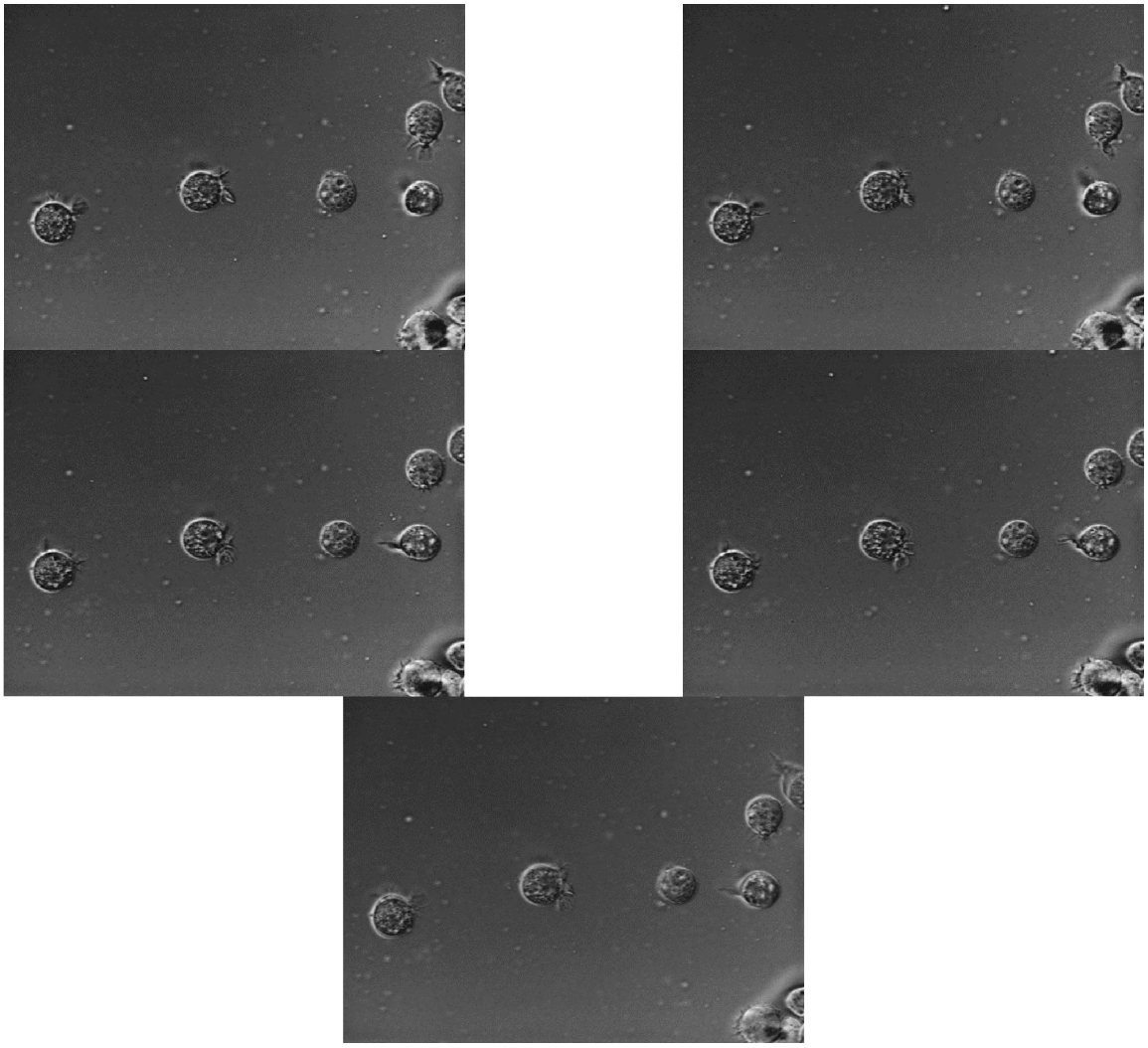
قسمت و:

شکل ۵-۲: سمت راست، تصاویر حاصل با استفاده از تابع imcomplement متلب حاصل شده است که شرح نحوه‌ی عملکرد آن در قسمت قبل قید گردید.

همانطور که قابل مشاهده است در تصاویر نهائی با توجه به غالب‌بودن نواحی مشکی، جزئیات سفید یا خاکستری تعبیه‌شده در نواحی تیره‌ی یک عکس تقویت شده‌اند و این مسئله دید بهتری به بیننده برای تحلیل عکس می‌دهد.

قسمت ز:



شکل ۶-۲: چهار تصویر بالائی، تصاویر اولیه پیش از مخلوط‌شدن بوده و تصویر پایینی از مخلوط شدن این تصاویر طبق فرمول زیر حاصل شده است:

$$C = w_A A + w_B B + w_C C + w_D D$$

جواب سوال ۳

آشنائی با معکوس‌کردن تصاویر به صورت افقی و عمودی

شکل ۱-۳: تصویر ردیف بالائی تصویر اصلی بوده و در ردیف پایین، تصویر سمت چپ، حاصل از معکوس‌کردن قسمتی دل‌بخواهی از تصویر به صورت عمودی و تصویر سمت راست، حاصل از انجام همین عمل به صورت افقی می‌باشد.

جواب سوال ۴

آشنائی با دقت نمایش سطوح خاکستری

نمایش با دو بیت

نمایش با یک بیت

نمایش با چهار بیت

نمایش با سه بیت

نمایش با شش بیت

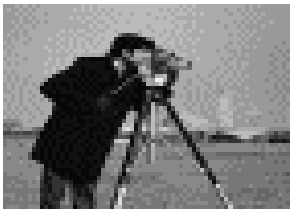
نمایش با پنج بیت

نمایش با هفت بیت

شکل ۱-۴: در تصاویر بالا هر کدام از آن‌ها با تنها تعدادی از هشت بیت شاخص سطوح روشنائی نمایش داده شده و سایر بیت‌ها مقدار صفر گرفته‌اند و همانطور که قابل مشاهده است هرچه تعداد بیت‌های نمایش بالا می‌رود وضوح تصویر نیز به دلیل افزایش سطوح روشنائی، بالا می‌رود.

جواب سوال ۵

آشنائی با نحوه‌ی تغییر اندازه‌ی یک تصویر به ابعاد کوچکتر



شکل ۱-۵: در اینجا تصاویر سمت راست تغییریافته‌ی تصاویر سمت چپ به ابعاد کوچکتر می‌باشند.

در تابع نوشته‌شده برای این کار پس از دریافت ابعاد جدید کوچکتر از کاربر، نسبت ابعاد اولیه به ابعاد جدید را یافته و سپس برای محاسبه‌ی شدت روشنائی سطوح مختلف تصویر جدید (در اینجا RGB)، به هر پیکسل از هر سطح تصویر جدید، یک بلاک از سطح مربوطه‌ی تصویر اولیه اختصاص داده و از مقادیر تمامی پیکسل‌های این بلاک میانگین گرفته و به پیکسل متناظر در تصویر جدید اختصاص می‌دهیم و همانطور که قابل مشاهده است کیفیت تصویر حاصل نیز در جریان این عمل کاهش می‌یابد.

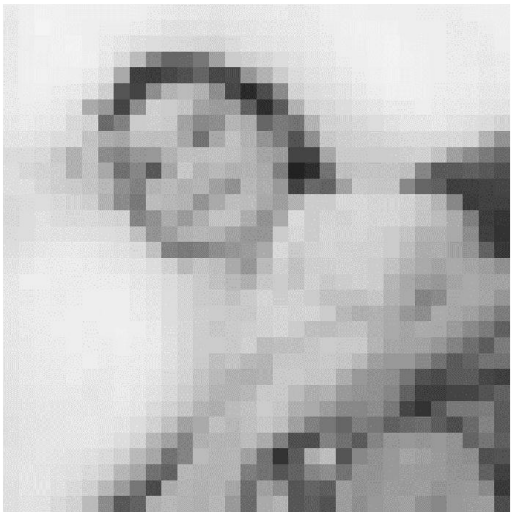
جواب سوال ۶

آشنائی با نحوه‌ی زِبر و خشن‌نمودن یک تصویر

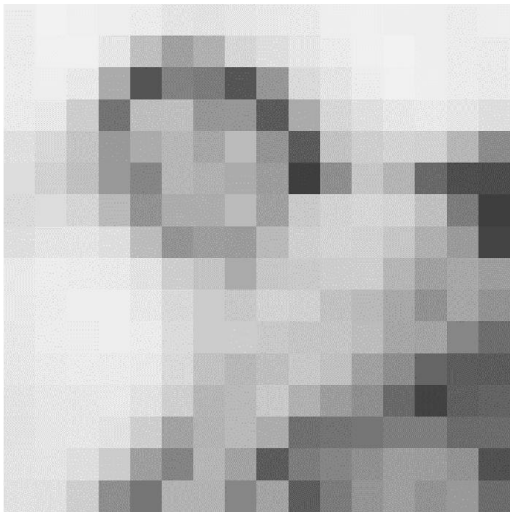
bSize = 4

bSize = 8

bSize = 16



bSize = 32



شکل ۱-۶: در تصاویر بالا در هر کدام کل تصویر به بلاک‌هایی مربعی با ابعاد قیدشده در بالای تصویر افراز شده و سپس مقادیر پیکسل‌های هر بلاک با مقدار میانگین کلیه‌ی پیکسل‌های هر بلاک جایگزین می‌گردد و در نتیجه‌ی این عمل ابعاد عکس حاصل بدون تغییر باقی می‌ماند.

همانطور که قابل مشاهده است هر چه که ابعاد بلاک مربوطه افزایش می‌یابد نمایش زِبرتر و خشن‌تری از تصویر مربوطه داریم، چرا که تعداد پیکسل‌های بیشتری از تصویر محو می‌شوند.

جواب سوال ۷

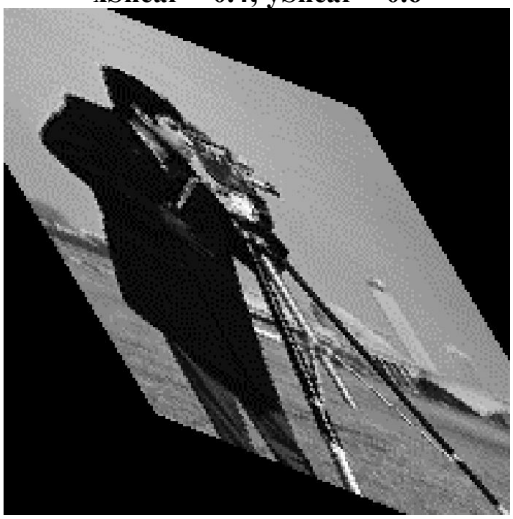
آشنائی با نحوه‌ی shear نمودن و یا کِش‌آوردن یک تصویر

تصویر اصلی

xShear = 0.2; yShear = 0.4

xShear = 0.3; yShear = 0.2

xShear = 0.4; yShear = 0.6



شکل ۱-۷: تصاویر بالا شامل تصویر اصلی و تصاویر shear شده در راستای محورهای x و y و حول مرکز عکس با مقادیر متفاوت xShear و yShear می‌باشند.

جواب سوال ۸

آشنائی با نحوه‌ی ترکیب‌نمودن دو تصویر

تصویر اول

تصویر دوم

شکل ۱-۸: در اینجا دو تصویر ورودی با ابعاد یکسان داریم که دارای یک بخش کاملا مشترک می‌باشند و برای یافتن آن بخش، در یک حلقه‌ی for با تعداد تکرار به اندازه‌ی تعداد ستون‌های یک تصویر، از منتهاالیه ستون‌های تصویر اول و ابتدای ستون‌های تصویر دوم شروع کرده و به اندازه‌ی مقدار شمارنده‌ی حلقه، ستون انتخاب نموده و در نتیجه دو ماتریس داریم که پس از محاسبه‌ی میزان همبستگی (correlation) میان درایه‌های این دو ماتریس با استفاده از تابع corr(.) متلب، مقدار همبستگی یا همان شباهت میان این دو بلاک را در یک بردار سطری (به طول همان تعداد ستون‌های یک تصویر) و در درایه‌ی متناظر با شمارنده‌ی حلقه ذخیره می‌نمائیم. در نهایت پس از یافتن اندیس بیشینه‌ی مقادیر این بردار، به مکان یا همان شماره‌ی ستونی که نشانگر آغاز بلاک یکسان در تصویر اول و نیز شماره‌ی ستونی که پایان این بلاک در تصویر دوم می‌باشد پی می‌بریم.

حال اگر تصاویر دارای ابعاد m*n بوده و تعداد ستون‌های بلاک یکسان نیز t باشد، ابعاد تصویر جدید به صورت m*(2*n-t) خواهد بود که از ستون 1 تا n-t متعلق به تصویر اول و از ستون n-t+1 تا 2*n-t متعلق به تصویر دوم می‌باشد.

جواب سوال ۹

آشنائی با نحوه‌ی تصویربرداری و ذخیره‌ی یک تصویر در یک دوربین دیجیتال

یک دوربین دیجیتال دوربینی است که تصاویر و ویدیوهای دیجیتال را به صورت دیجیتال یا رقمی رمزگذاری کرده و برای تکثیر مجدد، آن‌ها را ذخیره می‌نماید. دوربین‌های دیجیتال از یک سیستم نوری بهره می‌برند که به طور معمول از یک لنز با یک دیافراگم متغیر برای متمرکزکردن نور بر روی یک سخت‌افزار ضبط تصویر استفاده می‌کنند. دیافراگم و شاتر میزان صحیح نور ورودی به سخت‌افزار ضبط‌کننده را مانند حالت فیلم‌برداری تنظیم می‌نمایند با این تفاوت که در اینجا سخت‌افزار ضبط‌کننده‌ی تصویر، الکترونیکی می‌باشد نه شیمیائی.

حسگرهای نوری از هزاران ردیف المان نیمه‌هادی بسیار کوچک و حساس به نور تشکیل شده‌اند که می‌توانند ذرات یا فوتون‌های نور را به بار الکتریکی تبدیل کنند. حال هر چه شدت نور ورودی بیشتر یا کمتر باشد، الکتریسیته ایجاد شده متعاقباً دست‌خوش تغییر می‌شود. جنس این صفحه‌های حسگر اغلب از عناصری از جمله سیلیسیم و ژرمانیوم است. دو نوع عمده از حسگرهای تصویر دیجیتال عبارتند از: CCD و CMOS. یک حسگر CCD تنها یک تقویت‌کننده برای همگی پیکسل‌ها دارد، در حالی که در یک حسگر CMOS به ازای هر پیکسل یک تقویت‌کننده‌ی مخصوص وجود دارد. حسگرهای CMOS نسبت به حسگرهای از نوع CCD میزان انرژی کمتری مصرف می‌کنند. به طور نمونه شرکت کانن در دوربین‌های SLRخود تاکنون تنها از سنسورهای CMOS استفاده کرده است، در حالی که شرکت نیکون از هر دو نوع سنسور بهره می‌گیرد. دوربین‌های با یک حسگر کوچک از یک حسگر back-side-illuminated CMOS (BSI-CMOS) استفاده می‌کنند. کیفیت تصویر نهائی بیشتر بر قابلیت پردازش تصویر دوربین دیجیتال متکی هستند تا نوع حسگر مورد استفاده در دوربین مربوطه.

از جمله انواع فرمت‌های ذخیره‌سازی تصویر در دوربین‌های دیجیتال، فرمت‌های RAW ، TIFF و JPEG می‌باشند. هر دوی فرمت‌های RAW و TIFF هیچ‌گونه فشرده‌سازی به تصویر حاصله جهت ذخیره‌ی فضای اضافی بر روی کارت حافظه، اعمال نمی‌نمایند. زمانی که دوربین دیجیتال یک تصویر رقمی را با فرمت RAW و یا TIFF ذخیره می‌نماید، تصویر حاصله تمامی اطلاعات ضبط‌شده توسط حسگر تصویر دوربین را شامل می‌شود. فرمت JPEG بیش از آن است که یک فرمت ذخیره‌سازی تصویر معمولی به شمار آید و البته که عمل فشرده‌سازی را روی تصویر نهائی انجام می‌دهد. از جمله دلایل محبوبیت این نوع فرمت ذخیره‌سازی تصویر، قابلیت آن در امکان ذخیره‌سازی شمار بسیاری تصاویر در یک کارت حافظه فقط ۱۲۸ مگابایتی است.