





دانشکده مهندسی کامپیوتر و فن آوری اطلاعات
گروه هوش مصنوعی

بکارگیری راهکارهای مناسب پیش پردازش تصویر برای عملکرد بهتر تکنیک‌های فشرده‌سازی تصاویر

زهرا قیسی

استاد راهنما:

دکتر حمید حسن پور

استاد مشاور:

مهندس سکینه اسدی امیری

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۳

دانشگاه شاهرود

دانشکده: مهندسی کامپیوتر و فن آوری اطلاعات

گروه: هوش مصنوعی

پایان نامه کارشناسی ارشد خانم زهرا قیسی

تحت عنوان: بکارگیری راهکارهای مناسب پیش پردازش تصویر به منظور عملکرد بهتر تکنیک‌های فشرده‌سازی تصویر

در تاریخ _____ توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه _____ مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی: مهندس سکینه اسدی		نام و نام خانوادگی: پروفسور حمید حسن پور

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی:		نام و نام خانوادگی:
			نام و نام خانوادگی:

تقدیم به...

... او که ذخیره‌ای است برای بازنمایدن فریضه‌ها و سنت‌های خداوند در عالم.

تقدیر و تشکر

حال که به لطف و رحمت حضرت حق، مراحل این پایان نامه روبه اتمام نهاده، بر خود لازم می دانم تا از همه کسانی که در پیشبرد اهداف این پایان نامه بنده رایاری نمودند، سپاس و قدردانی به عمل آورم.

ابتدا از زحمات و پشتیبانی بی دریغ و بی سائبه استاد ارجمندم جناب آقای پروفور حسن پور که راهنمایی این تحقیق را بر عهده داشتند، کمال تشکر را دارم. بی شک بدون حمایت و هم فکری ایشان انجام این پایان نامه مقدور نبود.

و نیز سپاسگزارم از خانم مهندس سکینه اسدی که از راهنمایی و مساعدت اینجانب در انجام این پژوهش مضائقه نکردند. علاوه بر این، بر خود واجب می دانم از اساتید گران قدر جناب آقای دکتر زاهدی و جناب آقای دکتر پویان که سعادت شاگردی ایشان را در دوره کارشناسی ارشد داشته ام، قدردانی کرده و از خداوند متعال برای این دو بزرگوار موفقیت و بهروزی مسألت دارم.

در انتها از تمام اعضای خانواده و بهسر عزیزم که در کلیه مراحل زندگی پشتیبان من بوده اند و مهدی هایشان امید بخش انجام این رساله بود، با تمام خضوع قدردانی می نمایم.

تعهد نامه

اینجانب زهرا قیسی دانشجوی دوره کارشناسی ارشد رشته مهندسی کامپیوتر-گرایش هوش مصنوعی دانشکده مهندسی کامپیوتر و فن آوری اطلاعات دانشگاه شاهرود نویسنده پایان نامه " بکارگیری راهکارهای مناسب پیش پردازش تصویر برای عملکرد بهتر تکنیک های فشرده سازی تصاویر " تحت راهنمایی دکتر حمید حسن پور متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه شاهرود می باشد و مقالات مستخرج با نام « دانشگاه شاهرود » و یا « University of Shahrood » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافت های آن ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزارها و تجهیزات ساخته شده) متعلق به دانشگاه شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

امروزه به دلیل حجم زیاد اطلاعات دیجیتال، فشرده‌سازی تصاویر به عنوان یک نیاز اساسی در سیستم‌های بایگانی و ارسال تصاویر مطرح می‌شود. روش‌های زیادی برای افزایش نرخ فشرده‌سازی تصویر ایجاد شده است. اما همچنان افزایش نرخ فشرده‌سازی و حفظ کیفیت تصویر بازاریابی شده یک ضرورت محسوب می‌شود. هر یک از الگوریتم‌های فشرده‌سازی موجود بر اساس یک یا چند ویژگی اقدام به شناسایی افزونگی در تصویر و در نتیجه فشرده‌سازی آن می‌کنند. بر همین اساس، هر یک از روش‌های فشرده‌سازی، به ازای تصاویر مختلف، نرخ فشرده‌سازی یکسانی را ارائه نمی‌دهند. نرخ‌های فشرده‌سازی آنها ممکن است به عواملی چون تنوع سطوح خاکستری، هموار بودن تصویر و یکسان بودن مقادیر پیکسل‌های مجاور در تصویر بستگی داشته باشد. بر این اساس، در این تحقیق راهکارهایی ارائه می‌شود که به عنوان یک مرحله پیش‌پردازش بر روی تصویر اعمال شوند تا پارامترهای تاثیرگذار مربوطه در آن تصویر را جهت فشرده‌سازی تقویت نمایند.

در این پایان‌نامه چهار روش پیش‌پردازش ارائه شده است. پردازش اول بدون اتلاف بوده و در دو گام صورت می‌گیرد. در گام اول، تصویر به بلوک‌های بدون هم‌پوشانی تقسیم شده و بلوک‌ها بر اساس افزایش (کاهش) ضرایب DC، مرتب می‌شوند. در گام دوم، هشت تبدیل هندسی برگشت‌پذیر بر روی تصویر حاصل از گام اول صورت می‌گیرد. از بین تصاویر حاصل، تصویری که منجر به بهترین نتیجه برای فشرده‌سازی می‌شود، انتخاب می‌گردد. این پردازش باعث کاهش دامن‌های تغییرات، در نتایج حاصل از محاسبات JPEG شده و نرخ فشرده‌سازی آن را بهبود می‌دهد. در روش دوم، سطوح خاکستری که از نظر مقدار، به هم نزدیک می‌باشند، در یک دسته قرار می‌گیرند. سپس، یک سطح خاکستری به عنوان نماینده برای هر دسته در نظر گرفته می‌شود. مقدار هر سطح خاکستری در تصویر اصلی با مقدار نماینده‌ی مربوط به دسته‌ی خودش، جایگزین می‌شود. دسته‌بندی و انتخاب نماینده به نحوی انجام شده است که خرابی تصویر حداقل باشد. در گام بعد، هیستوگرام تصویر پردازش شده، متراکم می‌شود. این پیش‌پردازش، با حفظ کیفیت تصویر، توانسته نرخ فشرده‌سازی الگوریتم Lossless JPEG را بهبود دهد. روش سوم مشابه روش دوم می‌باشد با این تفاوت که تصویر در ابعاد ثابت بلوک‌بندی شده و مقدار پارامتر تعیین‌کننده‌ی تعداد سطوح هر دسته، برای هر بلوک، آگاهانه‌تر و بر اساس سیستم بینایی انسان انتخاب می‌گردد. این پیش‌پردازش باعث بهبود نرخ فشرده‌سازی الگوریتم‌های JPEG2000 و Lossless JPEG شده است. در روش چهارم، با استفاده از الگوریتم ژنتیک فضای رنگ مناسبی برای فشرده‌سازی هر تصویر به طور مجزا ایجاد می‌گردد. این پیش‌پردازش منجر به بهبود عملکرد الگوریتم CIC شده است.

کلمات کلیدی:

فشرده‌سازی تصویر، افزونگی، پیش‌پردازش، بهبود نرخ فشرده‌سازی، هیستوگرام تصویر، Lossless JPEG، سیستم بینایی انسان، JPEG2000، فضای رنگ، الگوریتم ژنتیک.

لیست مقالات مستخرج از پایان نامه

[1] H. Hassanpour, Z. Gheisi and S. Asadi-Amiri, "A novel pre-processing approach for improving compression rate of JPEG2000", Journal of Information Systems and Telecommunication [\(JIST\)](#), 2014.

[۲] قیسی ز، حسن پور ح، اسدی س، (۱۳۹۳)، "ارائه روش پیش پردازش تصویر به منظور عملکرد بهتر تکنیک فشرده سازی هافمن"، ششمین کنفرانس فناوری اطلاعات و دانش، شاهرود، ایران.

فهرست مطالب

۱- مقدمه‌ای بر فشرده‌سازی تصویر	۱
۱-۱- مقدمه	۲
۲-۱- ضرورت و نقش پیش‌پردازش در فشرده‌سازی	۲
۳-۱- انواع افزودگی در تصویر و کاربرد آن در فشرده‌سازی	۳
۱-۳-۱- افزودگی کدینگ	۳
۲-۳-۱- افزودگی درون پیکسلی	۴
۳-۳-۱- افزودگی دیداری	۴
۴-۱- ساختار پایان‌نامه	۵
۵-۱- نتیجه‌گیری	۶
۲- تکنیک‌های فشرده‌سازی تصویر	۷
۱-۲- مقدمه	۸
۲-۲- مروری بر چند تکنیک فشرده‌سازی بدون اتلاف	۹
۱-۲-۲- رمزگذاری لایه‌های تک بیتی	۹
۲-۲-۲- رمزگذاری طول اجرا	۱۰
۳-۲-۲- رمزگذاری LZW	۱۱
۴-۲-۲- کدگذاری هافمن	۱۲
۳-۲-۳- مروری بر چند تکنیک فشرده‌سازی با اتلاف	۱۳
۱-۳-۲- کدینگ تبدیلی	۱۴
۲-۳-۲- کدینگ پیش‌بینی‌کننده با اتلاف	۱۷
۳-۳-۲- کوانتیزاسیون برداری	۱۸
۴-۳-۲- فرکتال کدینگ	۱۹
۵-۳-۲- کدینگ برش بلاک	۲۱
۶-۳-۲- کدینگ زیرباند	۲۱
۴-۲- فرمت‌های ذخیره‌سازی تصویر	۲۲
۱-۴-۲- فرمت ذخیره‌سازی JPEG	۲۳
۲-۴-۲- فرمت ذخیره‌سازی Lossless JPEG	۲۵
۳-۴-۲- فرمت ذخیره‌سازی JPEG_LS	۲۶
۴-۴-۲- فرمت ذخیره‌سازی JPEG2000	۲۸
۵-۴-۲- فرمت ذخیره‌سازی GIF	۲۹

۳۰	۵-۲- ویژگی‌های یک فضای رنگ مناسب در فشرده‌سازی تصویر.....
۳۰	۶-۲- نتیجه‌گیری.....
۳۳	۳- مروری بر روش‌های پیشین پیش‌پردازش.....
۳۴	۱-۳- مقدمه.....
۳۴	۲-۳- مروری بر پژوهش‌های موجود.....
۴۰	۳-۳- نتیجه‌گیری.....
۴۳	۴- معرفی و ارزیابی روشهای پیشنهادی.....
۴۴	۱-۴- مقدمه.....
۴۴	۲-۴- نحوه‌ی بکارگیری ضرایب DC در JPEG.....
۴۵	۳-۴- جابجایی نواحی تصویر بر اساس ضرایب DC، به منظور کاهش دامنه‌ی تغییرات (روش پیشنهادی اول).....
۴۸	۱-۳-۴- مثالی از عملکرد روش پیشنهادی اول.....
۴۴-۴	کاهش تنوع سطوح خاکستری تصویر و متراکم نمودن هیستوگرام به منظور کاهش آنتروپی (روش پیشنهادی دوم)
۴۹	
۵۲	۱-۴-۴- مثالی از عملکرد روش پیشنهادی دوم.....
۵-۴	دسته‌بندی سطوح خاکستری و متراکم کردن هیستوگرام تصویر بر اساس سیستم بینایی انسان (روش پیشنهادی سوم).....
۵۲	
۵۳	۱-۵-۴- عملکرد سیستم بینایی انسان نسبت به تغییرات.....
۵۴	۲-۵-۴- انتخاب پارامتر تعیین‌کننده‌ی تعداد سطوح هر دسته و دسته‌بندی سطوح خاکستری.....
۵۶	۳-۵-۴- مثالی از عملکرد روش پیشنهادی سوم.....
۶۰-۴	یافتن فضای رنگ مناسب برای هر تصویر به منظور افزایش نرخ فشرده‌سازی (روش پیشنهادی چهارم).....
۶۱	۱-۶-۴- ساختار کروموزوم.....
۶۲	۲-۶-۴- تابع برازش.....
۶۳	۳-۶-۴- الگوریتم فشرده‌سازی مورد استفاده برای روش پیشنهادی چهارم.....
۶۴	۴-۶-۴- مثالی از عملکرد روش پیشنهادی چهارم.....
۶۷	۷-۴- ارزیابی روش‌های پیشنهادی.....
۶۷	۱-۷-۴- توصیف پایگاه داده USD.....
۶۸	۲-۷-۴- توصیف پایگاه داده Kodak.....
۶۹	۳-۷-۴- معیارهای سنجش الگوریتم‌های فشرده‌سازی.....
۷۲	۴-۷-۴- نتایج حاصل از روش پیشنهادی اول بر روی تصاویر USD.....
۷۵	۵-۷-۴- نتایج پیش‌پردازش تصویر با روش پیشنهادی دوم در بهبود عملکرد Lossless JPEG.....
۷۸	۶-۷-۴- نتایج پیش‌پردازش تصویر با روش پیشنهادی سوم در بهبود عملکرد Lossless JPEG و JPEG2000.....
۸۱	۷-۷-۴- نتایج روش پیشنهادی چهارم در بهبود عملکرد الگوریتم CIC.....

۸۲	۴-۷-۸- بررسی زمان پردازش روش‌های پیشنهادی.....
۸۲	۴-۸- مقایسه کارایی روش‌های پیشنهادی با روش‌های موجود.....
۸۳	۴-۹- نتیجه‌گیری.....
۸۵	۵- جمع‌بندی و پیشنهادات.....
۸۶	۵-۱- جمع‌بندی.....
۱۰۸	۵-۲- پیشنهادات برای ادامه‌ی کار.....
۹۱	۶- پیوست (۱).....
۹۵	۷- پیوست (۲).....
۱۱۳	۸- مراجع.....

فهرست جداول

جدول ۱-۲: هشت الگوی پیشگویی در روش Loosless JPEG.....	۲۶
جدول ۱-۳: خلاصه‌ای از عملکرد روش‌های پیش‌پردازش موجود.....	۴۱
جدول ۱-۴: عملکرد روش پیشنهادی چهارم بر روی کانال‌های چند نمونه از تصاویر.....	۶۶
جدول ۲-۴: نتایج روش پیشنهادی اول بر روی تصاویر پایگاه داده USD.....	۷۳
جدول ۳-۴: نتایج حاصل از اعمال روش پیشنهادی دوم بر روی تصاویر پایگاه داده USD.....	۷۶
جدول ۴-۴: نتایج روش پیشنهادی چهارم بر روی تصاویر پایگاه داده Kodak.....	۸۱
جدول ۵-۴: مقایسه بین روش پیشنهادی سوم با BWCA در بهبود عملکرد JPEG.....	۸۲
جدول پ-۱-۲: نتایج حاصل از اعمال روش پیشنهادی سوم در بهبود Lossless JPEG.....	۹۶
جدول پ-۲-۲: نتایج حاصل از پیش‌پردازش سوم در بهبود عملکرد JPEG2000 بر روی تصاویر USD.....	۱۰۶
جدول پ-۳-۲: زمان پردازش هر یک از روش‌های پیشنهادی بر روی تصاویر پایگاه داده USD.....	۱۰۷

فهرست شکل ها

- شکل ۱-۱: تفاوت حساسیت چشم انسان نسبت به خرابی در نواحی با بافت مختلف ۵
- شکل ۱-۲: مثالی از کدینگ لایه‌های تک بیتی ۱۰
- شکل ۲-۲: : مثالی از رمزگذاری طول اجرا برای عناصر یک تصویر باینری ۱۱
- شکل ۳-۲: نحوه‌ی ایجاد واژه‌نامه در الگوریتم LZW ۱۲
- شکل ۴-۲: دیاگرام روش فشرده‌سازی با اتلاف ۱۳
- شکل ۵-۲: ترکیب خطی برخی الگوهای پایه ۱۴
- شکل ۶-۲: مراحل پردازش یک بلوک با کدینگ تبدیلی ۱۴
- شکل ۷-۲: مولفه‌های اصلی تبدیل گسسته‌ی کسینوسی ۱۶
- شکل ۸-۲: تقریب یک تصویر با استفاده از تعداد مختلفی از ضرایب گسسته‌ی کسینوسی ۱۶
- شکل ۹-۲: مراحل کدگذاری مربوط به یک سیستم کدینگ پیش‌بینی‌کننده با اتلاف ۱۷
- شکل ۱۰-۲: مراحل کدگذاری مربوط به یک سیستم کدینگ پیش‌بینی‌کننده با اتلاف ۱۷
- شکل ۱۱-۲: مثالی از کدینگ کوانتیزاسیون برداری ۱۸
- شکل ۱۲-۲: مثالی از دی کدینگ کوانتیزاسیون برداری ۱۹
- شکل ۱۳-۲: نمایش دو بلوک خود متشابه در تصویر ۲۰
- شکل ۱۴-۲: مثالی از کدینگ برش بلاک ۲۱
- شکل ۱۵-۲: کدینگ و دی کدینگ زیرباند ۲۲
- شکل ۱۶-۲: حذف تفاوت نامحسوس رنگ‌های نزدیک به هم در تکنیک فشرده‌سازی JPEG ۲۳
- شکل ۱۷-۲: مراحل روش JPEG ۲۴
- شکل ۱۸-۲: مقایسه مقدار پیشگویی شده با مقدار واقعی پیکسل در روش DPCM ۲۵
- شکل ۱۹-۲: موقعیت همسایه‌های پیکسل X ۲۵
- شکل ۲۰-۲: موقعیت پیکسل X و پیکسل‌های همسایه ۲۶
- شکل ۲۱-۲: دیاگرام الگوریتم JPEG_LS ۲۷
- شکل ۲۲-۲: مراحل فشرده‌سازی در الگوریتم JPEG2000 ۲۸
- شکل ۲۳-۲: مراحل بازیابی در الگوریتم JPEG2000 ۲۸
- شکل ۲۴-۲: تصویری متشکل از سه رنگ ۲۹
- شکل ۱-۳: دیاگرام مربوط به فشرده‌سازی و بازیابی تصویر به همراه فاز پیش‌پردازش ۳۴
- شکل ۲-۳: مثالی از عملکرد روش پیشنهاد شده در مرجع [۴۳] ۳۷
- شکل ۳-۳: نمونه‌ای از گراف تعریف شده در مرجع [۴۶] ۳۸
- شکل ۱-۴: مقادیر یک بلوک از تصویر پس از اعمال تبدیل گسسته‌ی کسینوسی و کوانتایز کردن ضرایب ۴۵
- شکل ۲-۴: ضرایب DC دو بلوک مجاور ۴۵
- شکل ۳-۴: بلوک‌بندی و جابجایی بلوک‌ها بر اساس ضرایب DC ۴۶

- شکل ۴-۴: تغییر ابعاد تصویر هنگام چرخش با درجات نامناسب ۴۷
- شکل ۵-۴: اعمال تبدیلات هندسی بر روی تصویر ۴۷
- شکل ۶-۴: نمایش جهت تفاضل ضرایب DC ۴۸
- شکل ۷-۴: نمونه‌ای از اعمال روش پیشنهادی ۴۸
- شکل ۸-۴: تصاویر حاصل از اعمال تبدیلات هندسی بر روی تصویر پردازش شده ۴۹
- شکل ۹-۴: تبدیل هیستوگرام تصویر اصلی به هیستوگرام خلوت ۵۰
- شکل ۱۰-۴: بلوک دیاگرام روش پیشنهادی دوم ۵۱
- شکل ۱۱-۴: مثالی از عملکرد روش پیشنهادی سوم ۵۲
- شکل ۱۲-۴: واکنش متفاوت چشم انسان نسبت به تغییرات در نواحی با بافت متفاوت ۵۳
- شکل ۱۳-۴: تاثیر پدیده‌ی پوشش در کاهش قابلیت رویت خرابی تصویر ۵۴
- شکل ۱۴-۴: حساسیت کم چشم نسبت به خرابی در نواحی بسیار روشن و بسیار تیره ۵۴
- شکل ۱۵-۴: تصاویر در بردارنده‌ی اطلاعات لازم برای بازیابی ۵۶
- شکل ۱۶-۴: بلوک‌های 256×256 تصویر اصلی به همراه هیستوگرام هر بلوک ۵۷
- شکل ۱۷-۴: بلوک‌های تصویر پردازش شده به همراه هیستوگرام هر بلوک ۵۸
- شکل ۱۸-۴: بلوک‌های تصویر با مقادیر سطوح خاکستری نزدیک به هم به همراه هیستوگرام هر بلوک ۵۹
- شکل ۱۹-۴: تصاویر در بردارنده‌ی اطلاعات لازم برای بازیابی ۶۰
- شکل ۲۰-۴: نمونه‌ای از اعمال روش پیشنهادی چهارم بر روی تصویر pepper ۶۰
- شکل ۲۱-۴: ساختار کروموزوم در تکنیک پیشنهادی ۶۱
- شکل ۲۲-۴: مراحل فشرده‌سازی CIC ۶۴
- شکل ۲۳-۴: تغییر فضای رنگ تصاویر با استفاده از روش پیشنهادی چهارم ۶۵
- شکل ۲۴-۴: چهار نمونه تصویر از تصاویر پایگاه داده USD ۶۸
- شکل ۲۵-۴: تصاویر پایگاه داده Kodak ۶۸
- شکل ۲۶-۴: بهبود عملکرد JPEG با استفاده از روش پیشنهادی اول ۷۴
- شکل ۲۷-۴: بهبود عملکرد lossless JPEG با استفاده از روش پیشنهادی دوم ۷۵
- شکل ۲۸-۴: مقایسه عملکرد lossless JPEG با روش پیشنهادی دوم و سوم بر روی تصاویر USD ۷۹
- شکل ۲۹-۴: بهبود عملکرد JPEG2000 با استفاده از روش پیشنهادی سوم ۸۰

فصل اول

مقدمه‌ای بر فشرده‌سازی تصویر

۱-۱- مقدمه

امروزه روند رو به رشد استفاده از سیستم‌های چندرسانه‌ای باعث شده تا تصویر و علوم مربوط به آن بیش از پیش مورد توجه قرار گیرد. بخش عمده‌ای از پیشرفت در زمینه‌های مختلف از جمله پزشکی، هوا و فضا، ساخت سیستم‌های ماهواره‌ای، سیستم‌های پزشکی و بسیاری زمینه‌های دیگر حاصل بکارگیری تکنیک‌های پردازش تصویر است. علاوه بر این امروزه تصویر یک عنصر سازنده در شبکه‌ی جهانی اینترنت است و صفحات وب، بدون تصویر کارایی کمتری دارند.

این کاربردها باعث شده که دانش پردازش تصویر یکی از مهمترین زمینه‌های کاری محققان شود و به شاخه‌های تخصصی‌تری تقسیم گردد. یکی از مهمترین این شاخه‌ها، فشرده‌سازی است. در فشرده‌سازی تصویر سعی می‌شود که تا حد امکان حجم اطلاعات مورد نیاز برای بیان تصویر، بدون از دست رفتن کیفیت کاهش یابد [۱].

فشرده‌سازی اطلاعات، امروزه در اکثر حوزه‌ها مورد استفاده قرار می‌گیرد و از تکنیک‌های آن برای فشرده‌سازی انواع داده‌ها از قبیل ویدئو، تصویر، صوت و متن استفاده می‌شود [۱]. در ادامه به چند مورد از مزایای فشرده‌سازی اشاره می‌شود:

- ✓ استفاده هر چه بهتر از پهنای باند خطوط انتقال
- ✓ کاهش زمان و هزینه در ارسال تصاویر
- ✓ ذخیره‌سازی در فضای کمتر و کاهش هزینه‌های سنگین منابع ذخیره‌سازی
- ✓ افزایش امنیت داده‌ی ارسالی در تکنیک‌های مخفی‌سازی اطلاعات

۱-۲- ضرورت و نقش پیش‌پردازش در فشرده‌سازی

گاهی یک الگوریتم فشرده‌سازی برای یک تصویر نرخ فشرده‌سازی بالایی را حاصل می‌کند، در حالی که یک تصویر دیگر توسط همین الگوریتم ممکن است با نرخ بالایی فشرده نشود. علت این است که هر الگوریتم فشرده‌سازی، با در نظر گرفتن پارامترهای خاصی عمل فشرده‌سازی را انجام می‌دهد و

نسبت به این پارامترها حساس است. متفاوت بودن این پارامترها در تصاویر مختلف باعث می‌شود که کارایی الگوریتم فشرده‌سازی نیز متفاوت شود. به عنوان مثال وقتی یک تصویر تنوع رنگ کمی داشته باشد، توسط الگوریتم هافمن با نرخ بسیار بالایی فشرده می‌شود حال آنکه همین الگوریتم، در فشرده کردن یک تصویر با تنوع رنگ بالا چندان موفق نخواهد بود. لذا اگر تصویر قبل از فشرده‌سازی به گونه‌ای پردازش شود که پارامترهای تاثیرگذار در فشرده‌سازی را تقویت کند، عملکرد الگوریتم فشرده‌سازی را بهبود خواهد داد. این امر ضرورت پیش‌پردازش را در فشرده‌سازی تصویر نشان می‌دهد. به همین دلیل، پیش‌پردازش تصویر را به عنوان یک مرحله‌ی مهم قبل از فشرده‌سازی مورد توجه قرار داده و در این پژوهش، پیش‌پردازش‌هایی را برای برخی الگوریتم‌ها پیشنهاد دادیم.

۱-۳- انواع افزونگی^۱ در تصویر و کاربرد آن در فشرده‌سازی

در تصویر انواع افزونگی وجود دارد. وجود همین افزونگی‌ها امکان فشرده‌سازی را فراهم می‌آورد. کاهش یک یا چند افزونگی از تصویر، مهم‌ترین بخش در الگوریتم‌های فشرده‌سازی است. هدف از کاهش افزونگی، حذف داده‌های تکراری از داده‌ی اصلی می‌باشد. به‌طور کلی سه نوع افزونگی در تصویر ممکن است وجود داشته باشد [۲]. در ادامه به توضیح هر یک، خواهیم پرداخت.

۱-۳-۱- افزونگی کدینگ^۲

هر قالب نمایشی تعداد بیت معینی را برای نمایش یک عنصر از تصویر در نظر می‌گیرد. به عنوان مثال اگر یک تصویر در حالت سطح خاکستری نمایش داده شود، برای تمامی سطوح خاکستری هشت بیت در نظر گرفته می‌شود. در حالیکه برخی از سطوح خاکستری می‌توانند در تعداد بیت کمتری نمایش داده شوند. برخی از الگوریتم‌های فشرده‌سازی سعی در کاهش این نوع افزونگی دارند. در این الگوریتم‌ها، تعداد بیت کمتر برای نمایش سطوح خاکستری که دارای احتمال وقوع بیشتری هستند و

¹ Redundancy

² Redundancy Coding

تعداد بیت بیشتر برای نمایش سطوح خاکستری که دارای احتمال وقوع کمتری هستند در نظر گرفته می‌شود. تکنیک‌های فشرده‌سازی که به این صورت عمل می‌کنند، با نام کدینگ با طول متغیر^۱ شناخته می‌شوند.

۱-۳-۲- افزونگی درون پیکسلی^۲

یک نوع مهم از انواع افزونگی‌ها، افزونگی درون پیکسلی است. در یک تصویر وابستگی بین یک پیکسل با پیکسل مجاورش بسیار بالاست. به همین علت، مقدار هر یک از پیکسل‌های داده شده می‌تواند به طور منطقی از پیکسل‌های همسایه‌اش پیش‌بینی شود. بنابراین می‌توان بخشی از اطلاعات یک پیکسل را (به دلیل امکان پیش‌بینی از پیکسل مجاور) نادیده گرفت. اطلاعات حاصل از الگوریتم‌هایی که بر اساس کاهش این نوع افزونگی عمل می‌کنند، قابل دید توسط انسان نمی‌باشند. در واقع یک تصویر آرایه‌ای دو بعدی از پیکسل‌هاست که معمولاً برای دید انسان به کار می‌رود. اما پس از حذف افزونگی درون پیکسلی، حاوی اطلاعات دیداری از نظر چشم انسان نیست. اطلاعات حاصل اگر قابل برگشت به اطلاعات اولیه باشند نگاشت انجام شده را بازگشت‌پذیر^۳ می‌نامند [۲].

۱-۳-۳- افزونگی دیداری^۴

چشم انسان نمی‌تواند به همه‌ی اطلاعات یک تصویر با یک میزان حساسیت واکنش نشان دهد. به عنوان مثال چشم انسان به خرابی در نواحی با بافت متراکم نسبت به نواحی با بافت یکنواخت، حساسیت کمتری نشان می‌دهد. این مطلب در شکل (۱-۱) نشان داده شده است. همانطور که مشخص است خرابی در نواحی با بافت یکنواخت، توسط چشم انسان کاملاً قابل تشخیص است.

^۱ Variable length Coding

^۲ Interpixel Redundancy

^۳ Reversible

^۴ Visual Redundancy



شکل ۱-۱: تفاوت حساسیت چشم انسان نسبت به خرابی در نواحی با بافت مختلف.

بنابراین در تصویر اطلاعاتی وجود دارند که چشم نسبت به آنها حساسیت کمی دارد. این گونه اطلاعات، افزونگی دیداری محسوب می‌شوند. چشم انسان مقدار روشنائی هر پیکسل را به‌طور کمی پردازش نمی‌کند، بلکه این سیستم، بیشتر خصوصياتی مثل لبه‌ها را مورد ارزیابی قرار می‌دهد پس در واقع برای توصیف یک جسم نیازی به همه‌ی اطلاعات موجود نیست.

الگوریتم‌هایی که سعی در حذف افزونگی دیداری دارند، اطلاعات با اهمیت بالا از دید چشم انسان را حفظ نموده و باقی اطلاعات را حذف می‌نمایند. بر این اساس بخشی از اطلاعات تصویر نادیده گرفته می‌شود در حالیکه آسیبی به کیفیت تصویر از دید انسان وارد نمی‌شود [۲].

۴-۱- ساختار پایان‌نامه

در این فصل، مفاهیم اولیه فشردسازی، ضرورت انجام فشردسازی و نقش پیش‌پردازش در بهبود کارایی الگوریتم‌های فشردسازی مطرح شد. از آنجا که پیش‌پردازش‌های پیشنهادی در این پژوهش بر اساس عملکرد الگوریتم‌های فشردسازی ارائه می‌شوند، لذا در فصل دوم به معرفی چند روش متداول فشردسازی و همچنین فرمت‌های رایج ذخیره‌سازی تصویر خواهیم پرداخت. تاکنون پژوهش‌هایی در زمینه‌ی بهبود عملکرد الگوریتم‌های فشردسازی انجام شده است، در فصل سوم این پژوهش‌ها مرور

خواهد شد. در فصل چهارم روش‌های پیش‌پردازش پیشنهادی و نحوه‌ی پیاده‌سازی هر یک از روش‌ها و نتایج آن‌ها مورد بحث و بررسی قرار می‌گیرد. در فصل پایانی نیز به جمع‌بندی و ارائه‌ی پیشنهادات برای ادامه‌ی کار خواهیم پرداخت.

۱-۵- نتیجه‌گیری

همانطور که مطرح شد، فشردگی‌سازی تصویر از پرکاربردترین علوم پردازش تصویر است. یکی از مهمترین مزیت‌های فشردگی‌سازی، ذخیره‌سازی اطلاعات در فضای کمتر و کاهش هزینه‌های سنگین منابع ذخیره‌سازی است.

امروزه، حجم اطلاعات دیجیتال روند رشد سریعی داشته و به همین دلیل، همواره سعی شده که این اطلاعات، با بیشترین نرخ فشردگی شوند و در عین حال کیفیت تصویر بازیابی شده حفظ گردد. پیش‌پردازش تصویر قبل از مرحله‌ی فشردگی‌سازی باعث می‌شود که عوامل تاثیرگذار در الگوریتم‌های فشردگی‌سازی تقویت شده و فشردگی‌سازی با نرخ بالاتری صورت پذیرد. لذا در این پایان‌نامه پیش‌پردازش‌هایی ارائه می‌شود که منجر به بهبود عملکرد چند الگوریتم فشردگی‌سازی گردد.

فصل دوم

تکنیک‌های فشرده‌سازی تصویر

۲-۱- مقدمه

روش‌های پیش‌پردازش پیشنهاد شده در این پایان‌نامه، بر اساس چگونگی عملکرد الگوریتم‌های فشرده‌سازی ارائه شده‌اند. از اینرو، در این بخش به معرفی تکنیک‌های پرکاربرد در فشرده‌سازی تصویر خواهیم پرداخت. به طور کلی تکنیک‌های موجود در فشرده‌سازی تصویر در دو دسته‌ی با اتلاف^۱ و بدون اتلاف^۲ قرار می‌گیرند [۳]. در روش‌های بدون اتلاف، فشرده‌سازی به گونه‌ای انجام می‌شود که هیچ اطلاعاتی از بین نمی‌رود. در نتیجه تصویر بازیابی شده از نظر مقداری برابر با تصویر اصلی است [۳]. این روش‌ها بدون نویز^۳ نیز نامیده می‌شوند، زیرا هیچ نویزی به تصویر اصلی اعمال نمی‌کنند. همچنین با نام آنتروپی کدینگ^۴ هم شناخته می‌شوند، زیرا از تکنیک‌های تجزیه^۵ برای مینیمم کردن افزونگی استفاده می‌کنند. الگوریتم‌های فشرده‌سازی با اتلاف در کوچک کردن حجم تصویر بسیار بهتر عمل می‌کنند. علت این است که این تکنیک‌ها در عمل بخشی از داده‌های اصلی تصویر را حذف می‌کنند. به همین دلیل تصویر بازیابی شده کاملاً برابر با تصویر اصلی نخواهد بود [۴].

در این فصل ابتدا به توضیح چند تکنیک پرکاربرد از روش‌های فشرده‌سازی با اتلاف و بدون اتلاف خواهیم پرداخت. سپس چگونگی عملکرد چند فرمت مهم ذخیره‌سازی تصویر را توضیح خواهیم داد. در پایان نیز ویژگی‌های یک فضای رنگ مناسب برای فشرده‌سازی مطرح خواهد شد.

¹ Lossy

² Lossless

³ Noiseless

⁴ Entropy Encoding

⁵ Decomposition

۲-۲- مروری بر چند تکنیک فشرده‌سازی بدون اتلاف

همانطور که در بخش (۱-۲) گفته شد در روش‌های فشرده‌سازی بدون اتلاف، هیچ اطلاعاتی از بین نمی‌رود. روش‌های رمزگذاری لایه‌های بیتی^۱، رمزگذاری هافمن، رمزگذاری طول اجرا^۲ و رمزگذاری LZW از جمله روش‌های بدون اتلاف فشرده‌سازی تصویر به‌شمار می‌روند [۵-۶]. در ادامه به توضیح هر یک از این روش‌ها خواهیم پرداخت.

۲-۲-۱- رمزگذاری لایه‌های تک بیتی

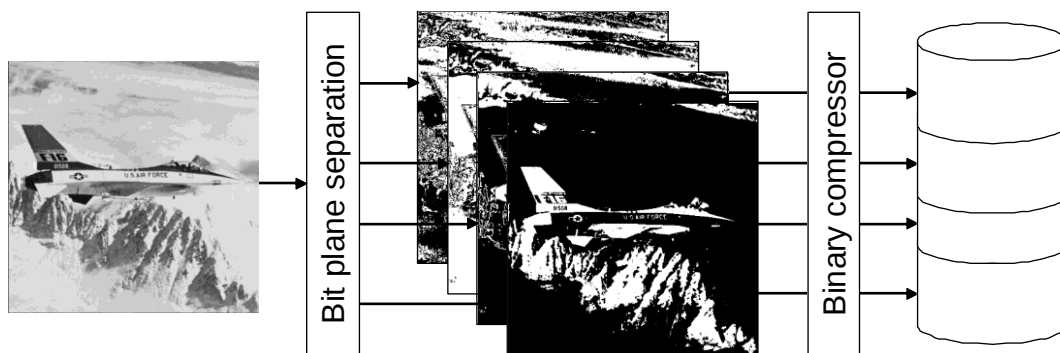
در رمزگذاری لایه‌های تک بیتی، ابتدا تصویری که شامل پیکسل‌هایی با هشت بیت است به هشت تصویر که دارای پیکسل‌هایی با یک بیت است، تجزیه می‌شود [۷]. فرمول (۱-۲) نحوه‌ی جداسازی تصاویر را به لایه‌های تک بیتی نشان می‌دهد.

$$x_i = \begin{cases} \text{mod}(x, 2^i) & i = 7 \\ \text{mod}(x - \sum_{j=i+1}^7 x_j 2^j, 2^i) & i = 0, 1, \dots, 6 \end{cases} \quad (1-2)$$

پس از جداسازی، هر یک از لایه‌ها فقط شامل مقادیر باینری صفر و یک خواهند بود. سپس از یک فشرده‌ساز باینری بدون اتلاف، برای فشرده کردن داده‌های هر لایه استفاده می‌شود. فشرده‌ساز باینری، تکنیکی است که برای فشرده کردن داده‌های باینری بکار می‌رود. نمونه‌ای از این فشرده‌ساز در بخش (۲-۲-۲) توضیح داده خواهد شد. ایده‌ی اصلی این روش این است که با تجزیه تصویر به صفحات تک بیتی، افزونگی در تصاویر بیتی افزایش می‌یابد. همچنین پیکسل‌های مجاور، بخصوص در بیت‌های بالاتر با احتمال بیشتری مقدار یکسان دارند. در نتیجه امکان فشرده‌سازی فراهم می‌شود. شکل (۱-۲) نمایی از این کدینگ را نشان می‌دهد.

¹ Bitplan Encoding

² Run length Encoding

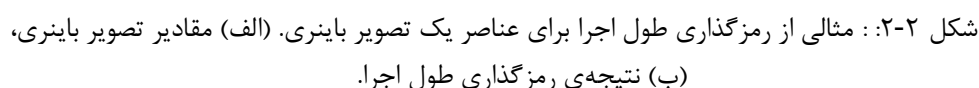


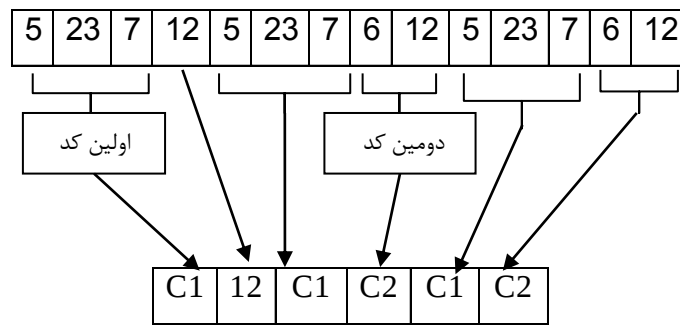
شکل ۱-۲: مثالی از کدینگ لایه‌های تک بیتی [۸].

۲-۲-۲- رمزگذاری طول اجرا

در این روش کدگذاری، اعضای تصویر با زوج (C_i, I_i) نگاشت می‌شوند. C_i بیانگر شدت روشنایی یا رنگ بوده و I_i معرف طول تکرار C_i در یک دنباله است. در رمزگذاری طول اجرا برای یک تصویر باینری، گروه پیوسته‌ی صفرها و یک‌ها برای هر سطر از تصویر شمارش می‌شود. به جای تکرار سمبل‌های یکسان و تکراری، تنها تعداد تکرار هر سمبل ذخیره می‌شود. این عمل برای هر سطر از تصویر انجام می‌شود. انتهای هر سطر نیز با یک سمبل مخصوص انتهای سطر مشخص می‌گردد. سپس طول تکرار هر عنصر، با استفاده از یک کدکننده‌ی بدون اتلاف مانند کدینگ هافمن کد می‌شود [۹-۱۰]. در شکل (۲-۲) مثالی از کدگذاری با این روش نشان داده شده است.

قابل ذکر است که در این روش میزان حجم لازم برای ذخیره‌سازی، به داده‌ی اولیه وابسته است. امکان دارد در بدترین حالت، حجم داده‌ی فشرده شده از حجم داده‌ی اصلی بیشتر شود. اما استفاده از این روش برای فشرده کردن تصاویری با حوزه‌ی وسیعی از رنگ‌های (شدت روشنایی‌های) بهم پیوسته می‌تواند منجر به نرخ فشرده‌سازی بالایی شود.





شکل ۳-۲: نحوه‌ی ایجاد واژه‌نامه در الگوریتم LZW.

۲-۲-۴ - کدگذاری هافمن

این تکنیک توسط دیوید هافمن در سال ۱۹۵۲ ایجاد شد و یکی از قدیمی‌ترین روش‌های فشرده‌سازی محسوب می‌شود. از این الگوریتم به عنوان یک فشرده‌ساز در یک مرحله از سایر روش‌های فشرده‌سازی نیز استفاده می‌شود [۱۴]. کدینگ هافمن، یک روش کدگذاری با طول متغیر و بدون اتلاف است [۱۴]. کدینگ با طول متغیر بدین معنی است که کد بیتی همه‌ی نمادها به یک اندازه نمی‌باشد. یعنی نمادهایی با فراوانی بیشتر، دارای کد هافمن کوتاهتری نسبت به نمادهایی با فراوانی کمتر می‌باشند.

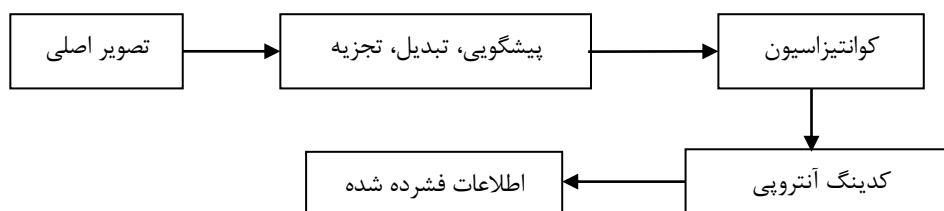
در الگوریتم هافمن، ابتدا داده‌ها بر اساس تعداد فراوانی‌شان مرتب می‌شوند، سپس درخت هافمن ایجاد می‌گردد. درخت هافمن یک درخت دودویی است که در آن، تمام گره‌ها غیر از برگ‌ها دارای دو فرزند هستند. یک گره جدید از ترکیب دو گره با کمترین نرخ فراوانی ایجاد می‌شود. (نرخ فراوانی این گره برابر با مجموع نرخ فراوانی دو گره ترکیب شده، می‌باشد). این گره جدید جایگزین دو گره قبلی می‌شود. این روند تا ایجاد کامل درخت، ادامه می‌یابد. برای اختصاص کد، از رأس درخت شروع به حرکت نموده و به هر کدام از شاخه‌های سمت چپ شماره یک و به شاخه‌های سمت راست، شماره صفر اختصاص داده می‌شود. سپس کد مربوط به هر داده با پیمایش از رأس تا آن داده‌ی مربوطه به دست می‌آید.

در رمزگذاری هافمن هیچ کدی پیشنهاد نمی‌دهد. این ویژگی باعث شده تا در مرحله‌ی کدگذاری ابهامی ایجاد نشود. قابل ذکر است که در رمزگذاری هافمن کم بودن تنوع داده‌ها در افزایش نرخ فشردگی موثر است [۱۷-۱۵].

۳-۲- مروری بر چند تکنیک فشردگی با اتلاف

همانطور که گفته شد، تکنیک‌های فشردگی با اتلاف در هر بار فشردگی و بازیابی، باعث تنزل کیفیت تصویر می‌شوند. در شکل (۲-۴) دیاگرام روش فشردگی با اتلاف نشان داده شده است. مراحل پیشگویی، تبدیل و تجزیه به‌طور کامل برگشت‌پذیر هستند. در فاز کوانتیزاسیون بخشی از اطلاعات از بین می‌رود. فاز آنتروپی کدینگ نیز بدون از دست رفتن اطلاعات همراه است. در هنگام بازیابی نیز، عکس این مراحل انجام می‌شود [۱۸].

روشهای کدینگ تبدیلی^۱، کدینگ پیش‌بینی‌کننده با اتلاف، کدینگ کوانتیزاسیون برداری^۲، کدینگ فرکتال^۳، کدینگ برش بلوک^۴ و کدینگ زیرباند^۵ از جمله روش‌های فشردگی با اتلاف هستند. در ادامه هر یک از این روش‌ها توضیح داده خواهد شد.



شکل ۲-۴: دیاگرام روش فشردگی با اتلاف.

^۱ Transformation Encoding

^۲ Vector Quantization

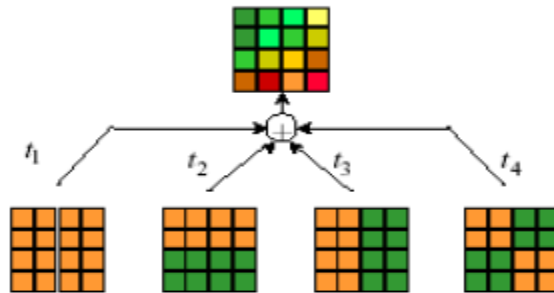
^۳ Fractal Coding

^۴ Block Truncation Coding

^۵ Subband Coding

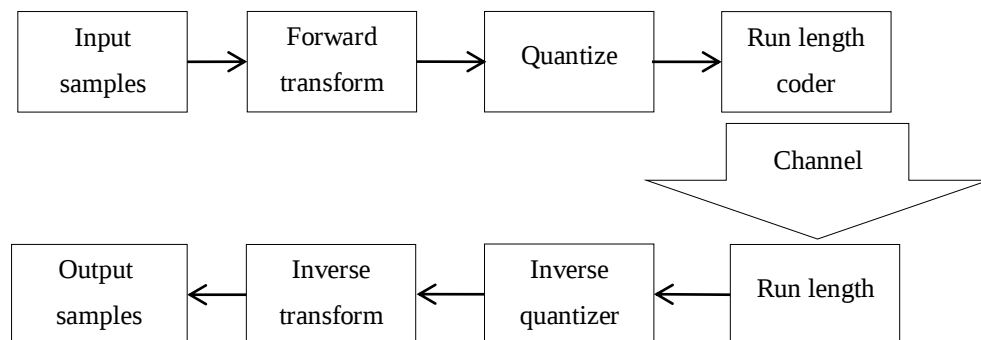
۲-۳-۱- کدینگ تبدیلی

در این نوع کدینگ، تصویر به بلوک‌های بدون هم‌پوشانی تقسیم شده و هر بلوک با مجموعه‌ای از الگوهای پایه نشان داده می‌شود. این الگوها به عنوان توابع پایه تبدیل شناخته می‌شوند. در شکل (۲-۵) یک بلوک از تصویر به صورت ترکیب خطی برخی از الگوهای پایه نشان داده شده است.



شکل ۲-۵: ترکیب خطی برخی الگوهای پایه.

شکل (۲-۶) مراحل پردازش یک بلوک را بر اساس کدینگ تبدیلی نشان می‌دهد. ابتدا تبدیل بر روی بلوک ورودی صورت می‌گیرد. سپس ضرایب تبدیل، کوانتیزه شده و کد می‌گردند. در مرحله کدگذاری نیز، بلوک اصلی از روی ضرایب کوانتیزه شده و از طریق معکوس تبدیل بازسازی می‌شوند.



شکل ۲-۶: مراحل پردازش یک بلوک با کدینگ تبدیلی.

تبدیل گسسته‌ی کسینوسی و تبدیل فوریه نیز نوعی کدینگ تبدیلی محسوب می‌شوند. هدف از تبدیلات ذکر شده این است که پیکسل‌های تصویر به ضرایبی در حوزه‌ی فرکانس تبدیل شده و با این عمل انرژی اطلاعات اصلی، در تعداد کمی از ضرایب متمرکز شوند. به این ضرایب، ضرایب معنادار گفته می‌شود. در کدینگ تبدیلی تنها ضرایب معنادار برای آنالیز یا کوانتیزاسیون مورد

استفاده قرار می‌گیرند و سایر ضرایب نادیده گرفته می‌شوند. از آنجا که تبدیل گسسته‌ی کسینوسی رایج‌تر و پرکاربردتر است، در ادامه به توضیح این تبدیل نیز می‌پردازیم.

۲-۳-۱-۱- تبدیل کسینوسی گسسته

استفاده از تبدیل کسینوسی گسسته، بسیار رایج است. این تبدیل، تصویر را به صورت حاصلجمع تعدادی تابع کسینوسی (سینوسی) با دامنه‌ها و فرکانس‌های مختلف بازنمایی می‌کند [۱۹]. بردارهای پایه یک بعدی n نقطه‌ای این تبدیل به صورت فرمول (۲-۱) تعریف می‌شوند.

$$h_k(n) = \alpha(k) \cos((2n+1)k\pi/2N) \quad (۲-۱)$$

که در این رابطه $\alpha(k)$ به صورت فرمول (۲-۲) جایگزین می‌شود.

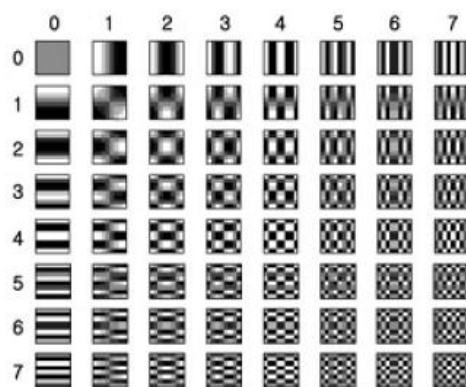
$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0 \\ \sqrt{\frac{2}{N}} & k = 1, 2, \dots, N-1 \end{cases} \quad (۲-۲)$$

تبدیل و معکوس تبدیل به ترتیب بر اساس فرمول‌های (۲-۳) و (۲-۴) به دست می‌آیند.

$$h_n(k) = \alpha(k) \sum_{n=0}^{N-1} f(n) \cos \frac{(2n+1)k\pi}{2N} \quad (۳-۲)$$

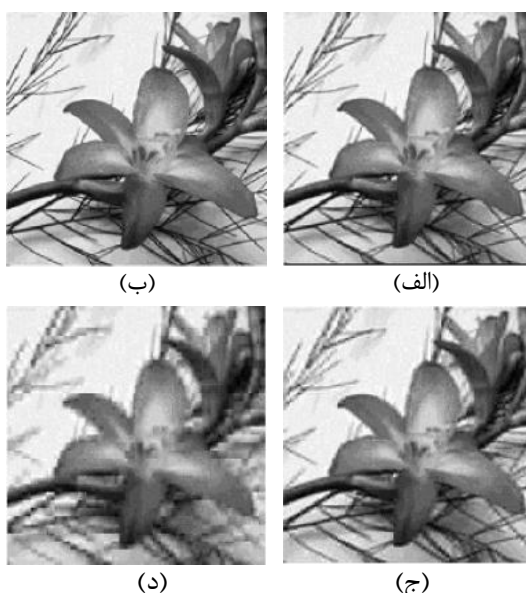
$$h_k(n) = \sum_{N=0}^{N-1} \alpha(k) t(n) \cos \frac{(2n+1)k\pi}{2N} \quad (۴-۲)$$

برای به دست آوردن تبدیل کسینوسی گسسته دو بعدی، در یک بلوک از تصویر، می‌توان ابتدا تبدیل یک بعدی را بر روی هر ردیف از این بلوک انجام داد و سپس بر روی هر ستون از بلوک حاصل، بار دیگر تبدیل را اعمال کرد. شکل (۲-۷) مولفه‌های اصلی برای تبدیل گسسته‌ی کسینوسی در ابعاد 8×8 را نشان می‌دهد. مولفه‌ی $D(0,0)$ ، ضریب DC و بیانگر میانگین تصویر می‌باشد. سایر ضرایب، مقادیر AC هستند که بیان‌کننده‌ی میزان تغییرات مقادیر سطح خاکستری در جهت و نرخ مشخص می‌باشند [۲۰].



شکل ۷-۲: مولفه‌های اصلی تبدیل گسسته‌ی کسینوسی.

مقادیر شدت روشنایی در یک تصویر، معمولاً به صورت هموار تغییر می‌کنند. همچنین اطلاعات کلی تصویر در مولفه‌های فرکانس پایین و جزئیات تصویر در مولفه‌های فرکانس بالا قرار دارند. به همین دلیل، با استفاده از ضرایب تبدیل گسسته‌ی کسینوسی با فرکانس پایین می‌توان تقریب دقیقی از تصویر به‌دست آورد. در شکل (۸-۲)، تقریب یک تصویر با استفاده از تعداد مختلفی از ضرایب، نشان داده شده است. همانطور که در شکل (۸-۲) مشخص است، تنها با ۱۶ ضریب، از ۶۴ ضریب می‌توان بلوک اصلی را به‌خوبی نمایش داد.



شکل ۸-۲: تقریب یک تصویر با استفاده از تعداد مختلفی از ضرایب گسسته‌ی کسینوسی. (الف) شکل اصلی با تمام ضرایب، (ب) تقریب با ۱۶ ضریب، (ج) تقریب با هشت ضریب، (د) تقریب با چهار ضریب.

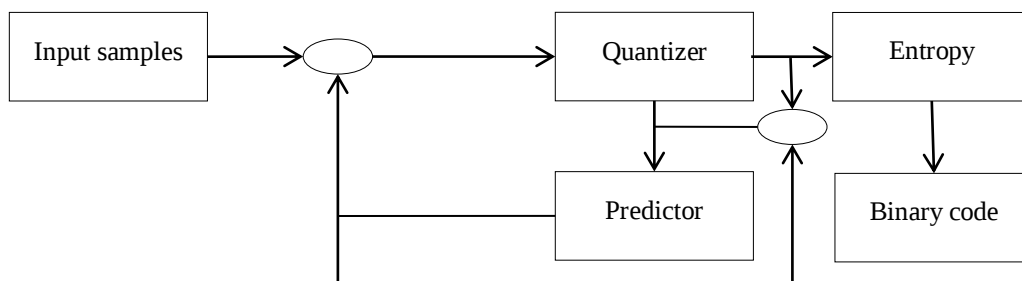
۲-۳-۲- کدینگ پیش‌بینی‌کننده با اتلاف

در یک تصویر، مقدار پیکسل حاضر، معمولاً تغییرات زیادی نسبت به پیکسل‌های مجاورش ندارد. به همین دلیل می‌توان مقدار یک پیکسل را از روی پیکسل‌های مجاورش پیش‌بینی نموده و تنها خطای پیش‌بینی را مشخص کرد. برای این امر می‌توان از کدینگ پیش‌بینی‌کننده استفاده نمود. کدینگ پیش‌بینی‌کننده می‌تواند به صورت با اتلاف یا بدون اتلاف باشد. در نوع با اتلاف، خطای پیش‌بینی ابتدا کوانتیزه شده و سپس کد می‌شود. اما در یک کدکننده‌ی بدون اتلاف، مرحله‌ی کوانتیزاسیون انجام نمی‌شود.

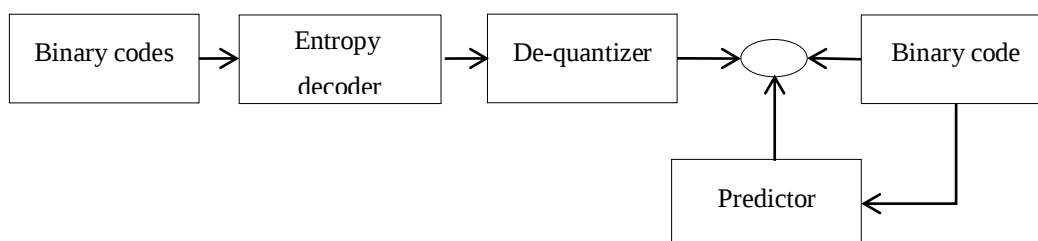
خطای پیش‌بینی عمدتاً در نزدیکی صفر متمرکز است و آنتروپی کمتری نسبت به نمونه‌های اصلی دارد. با یک کدکننده بدون اتلاف مثل هافمن، می‌توان مقادیر خطا را در تعداد بیت کمتری کد نمود.

شکل (۲-۹) مراحل کدگذاری و شکل (۲-۱۰) مراحل کدگشایی یک سیستم کدینگ پیش‌بینی-

کننده با اتلاف را نشان می‌دهد.



شکل ۲-۹: مراحل کدگذاری مربوط به یک سیستم کدینگ پیش‌بینی‌کننده با اتلاف.

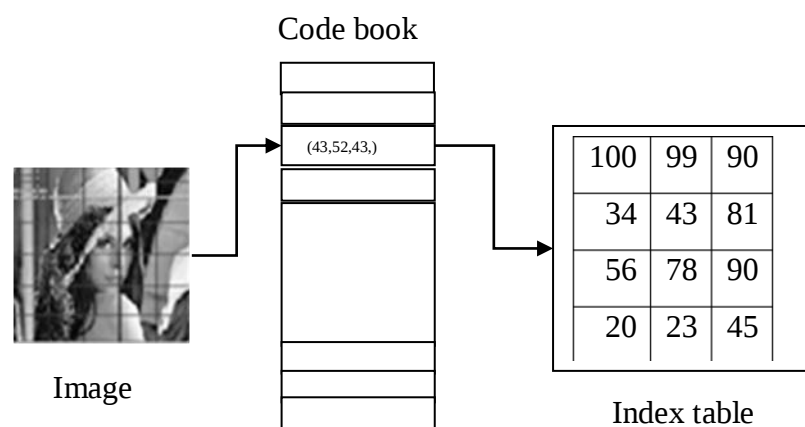


شکل ۲-۱۰: مراحل کدگشایی مربوط به یک سیستم کدینگ پیش‌بینی‌کننده با اتلاف.

۲-۳-۳- کوانتیزاسیون برداری

کوانتیزاسیون برداری، راهبرد دیگری برای فشرده‌سازی تصویر است. ایده‌ی این روش، تعیین بهترین دسته از الگوهای بلوک پایه است. الگوهای پایه باید به گونه‌ای انتخاب شوند که به بهترین وجه بلوک‌های تصویر را نمایش دهند. الگوهای پایه اصطلاحاً کلمه-کد^۱ و مجموعه‌ی تمام کلمه-کدها، کتاب-کد^۲ نامیده می‌شوند [۲۱-۲۲].

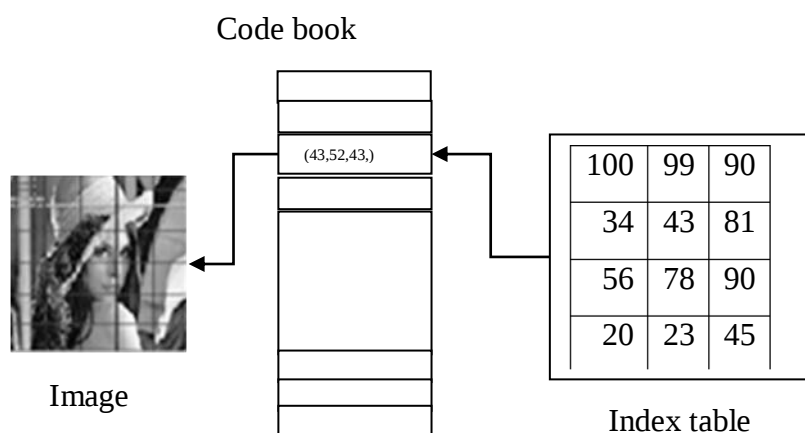
همانطور که در شکل (۲-۱۱) نشان داده شده است، در این روش ابتدا تصویر به بلوک‌های غیرهم-پوشان تقسیم می‌شود. هر بلوک به بردار تبدیل شده و این بردار به عنوان ورودی در نظر گرفته می‌شود. اندازه‌ی بردار، برابر با تعداد پیکسل‌های بلوک است. سپس در کتاب-کد، کلمه-کدی که به بهترین وجه ممکن، این بردار را تقریب می‌زند جستجو شده و بر این اساس اندیسی به بردار ورودی اختصاص داده می‌شود. در هنگام بازیابی نیز عکس این مراحل انجام می‌شود (شکل (۲-۱۲)).



شکل ۲-۱۱: مثالی از کدینگ کوانتیزاسیون برداری.

^۱ Code-Word

^۲ Code-Book



شکل ۲-۱۲: مثالی از دی‌کدینگ کوانتیزاسیون برداری.

۲-۳-۴- فرکتال کدینگ

ایده‌ی اصلی این کدینگ، تجزیه‌ی تصویر با استفاده از تکنیک‌هایی مثل تجزیه‌ی رنگ، آشکارسازی لبه، طیف و بافت است. تصویر به بخش‌هایی تقسیم شده و هر بخش در کتابخانه فرکتال جستجو می‌شود. کتابخانه شامل کدهایی است که سیستم تابع تکرار شونده (IFS)^۱ نامیده می‌شوند. یک IFS مجموعه‌ای از تبدیلات انقباضی است که طرحتی را به خودش نگاشت می‌دهد. منظور از تبدیلات انقباضی، آن دسته از تبدیلات هستند که با اعمال بر روی یک شکل، سبب نزدیک‌تر شدن اجزای آن به یکدیگر می‌شوند.

سیستم تابع تکرارشونده محلی (LIFS)^۲ کلاس وسیع‌تری را نسبت به IFS شامل می‌شود. IFS روی کل فضا تعریف می‌شود. اما LIFS روی زیرمجموعه‌های دلخواه از یک فضا تعریف می‌شود. علت استفاده از تابع تکرارشونده محلی به جای تابع تکرارشونده این است که برای بسیاری از شکل‌ها و تصاویر، بکارگیری IFS برای کدگذاری غیرممکن است زیرا الزاماً تصاویر از کپی‌هایی از خود تشکیل نشده‌اند. به عبارت دیگر، دستیابی به تبدیل‌هایی که یک تصویر را به خودش نگاشت بدهد برای اکثر تصاویر غیرعملی است [۲۳].

^۱ Iterated Function System

^۲ Local Iterated Function System

برای پیاده‌سازی این کدینگ، تصویر به بخش‌هایی به نام بلوک برد (R_i) تقسیم می‌شود. سپس برای هر R_i می‌بایست بلوکی یافت شود که پس از تبدیل، شباهت کافی با آن داشته باشد. بلوک‌های یافت شده، بلوک دامنه (D_i) نامیده می‌شوند. برای R_i و D_i معین، تبدیلی جستجو می‌شود که خطای بین R_i و D_i تبدیل یافته را به حد قابل قبولی برساند [۲۴-۲۵].

برای بررسی میزان تطابق بین دو بلوک، لازم است که از معیاری برای این امر استفاده شود. این معیار، فاصله‌ی اقلیدسی بین دو بلوک در نظر گرفته شده و به صورت فرمول (۵-۲) می‌باشد.

$$\delta(f, g)^n = \sum_{k=0}^n (f(x, y) - g(x, y))^n \quad (5-2)$$

در فرمول (۵-۲)، $f(x, y)$ یک پیکسل از بلوک اصلی، $g(x, y)$ یک پیکسل از بلوک تبدیل یافته و $\delta(f, g)$ فاصله‌ی اقلیدسی دو بلوک است.

در شکل (۱۳-۲) دو بلوک از تصویر که با هم تطابق دارند مشخص شده است. این کدینگ برای تصاویر که در آنها نظم و قاعده‌ی خاصی دیده می‌شود و خود متشابه هستند با نرخ فشرده‌سازی بالا عمل می‌کند.

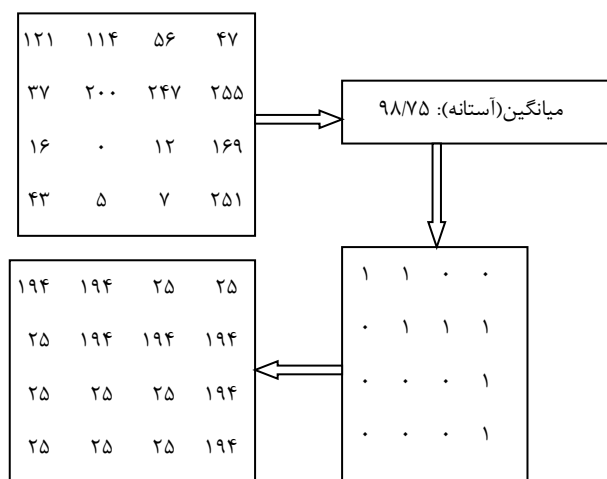


شکل ۱۳-۲: نمایش دو بلوک خود متشابه در تصویر. (الف) تصویر اصلی. (ب) محل دو بلوک خود متشابه در تصویر.

۲-۳-۵- کدینگ برش بلاک

در این تکنیک تصویر به بلوک‌هایی بدون هم‌پوشانی تقسیم می‌شود. برای هر بلوک یک مقدار آستانه و مقادیر بازسازی اتخاذ می‌گردد. مقدار آستانه معمولاً برابر با میانگین مقادیر پیکسل‌های همان بلوک در نظر گرفته می‌شود [۲۶]. سپس ماتریسی هم‌سایز با بلوک اصلی، ایجاد شده و اگر مقادیر عنصری در بلوک بزرگتر یا مساوی (کوچکتر) مقدار آستانه باشد، مقدار متناظرش در ماتریس جدید برابر با یک (صفر) خواهد شد. مثالی از این روش در شکل (۲-۱۴) نشان داده شده است. در مرحله‌ی بازیابی، برای هر گروه از مقادیر صفر یا یک در ماتریس، مقدار جدیدی قرار می‌گیرد. مقدار جدید برای مقادیر صفر برابر با میانگین پیکسل‌های متناظرشان در تصویر اصلی است. مقدار جدید برای مقادیر

یک نیز به همین ترتیب محاسبه می‌شود [۲۷].



شکل ۲-۱۴: مثالی از کدینگ برش بلاک.

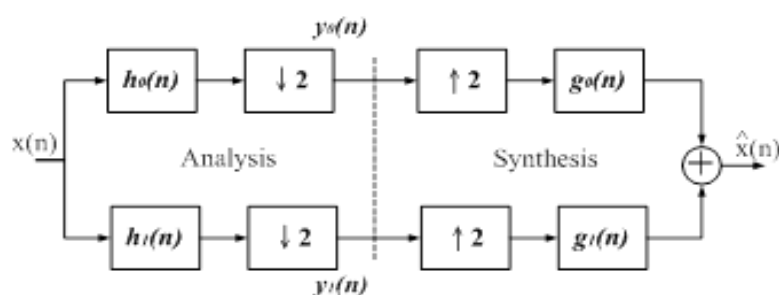
۲-۳-۶- کدینگ زیرباند

همانطور که در شکل (۲-۱۵) نشان داده شده است، در این روش، ابتدا تصویر با استفاده از فیلترها به دو زیرباند بالا و زیرباند پایین تجزیه می‌شود. پس از این مرحله، حجم داده‌ها دو برابر حجم داده‌ی

اصلی خواهد شد. لذا لازم است که عملیات کاهش بعد^۱ برای کاهش کمیت داده انجام شود. سپس عمل کوانتایز کردن و کد کردن برای هر زیرباند به صورت جداگانه انجام می‌شود.

در هنگام بازیابی، بر روی داده عمل افزایش بعد^۲ صورت می‌گیرد. پس از افزایش بعد، دو زیرباند با هم ترکیب شده تا تقریبی از داده‌ی اصلی به دست آید.

لازم به ذکر است که تقسیم تصویر به چند زیرباند، به تنهایی حجم را کاهش نمی‌دهد. سیستم ادراک انسان حساسیت متفاوتی نسبت به باندهای فرکانسی متفاوت دارد. به عنوان مثال سیستم بینایی انسان حساسیت کمتری نسبت به عناصر رنگ در باند فرکانس بالا دارد لذا از این مزیت استفاده می‌شود تا با تقسیم تصویر به چند زیرباند و حذف برخی داده‌ها، حجم تا حدی کاهش یابد و بر این اساس به میزانی از فشرده‌سازی دست یافت.



شکل ۲-۱۵: کدینگ و دی‌کدینگ زیرباند.

۲-۴- فرمت‌های ذخیره‌سازی تصویر

امروزه فرمت‌های مختلفی برای ذخیره کردن فایل‌های تصویری بکار گرفته می‌شود. هر فرمت ذخیره‌سازی کاربرد خاص خود را دارد. برخی از آنها تصویر را در کیفیت مطلوب ذخیره می‌کنند اما تصویر ذخیره شده فضای زیادی از حافظه را اشغال می‌کند. برخی دیگر بخشی از اطلاعات تصویر را حذف نموده و تصویر را در حجم بسیار کم ذخیره می‌کنند.

^۱ Down Sample

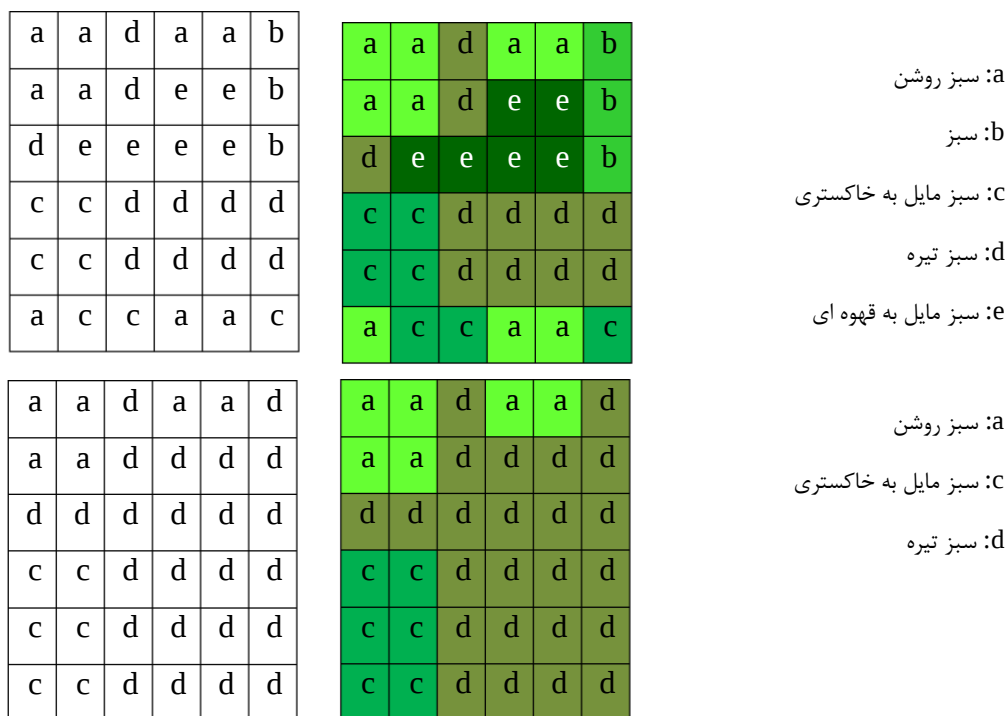
^۲ Up Sample

ممکن است در ذخیره‌سازی تصاویر دیجیتال از الگوهای فشرده‌سازی تصویر استفاده شود. لذا لازم است که در این بخش به توضیحاتی در مورد چند فرمت ذخیره‌سازی بپردازیم.

۲-۴-۱- فرمت ذخیره سازی JPEG

در این تکنیک، امکان ذخیره‌سازی تصویر به صورت با اتلاف و بدون اتلاف وجود دارد و در هر کدام، از دو روش کاملاً متفاوت استفاده می‌شود. در روش بدون اتلاف از پیشگویی کننده برای کاهش آنتروپی اطلاعات بهره گرفته شده است.

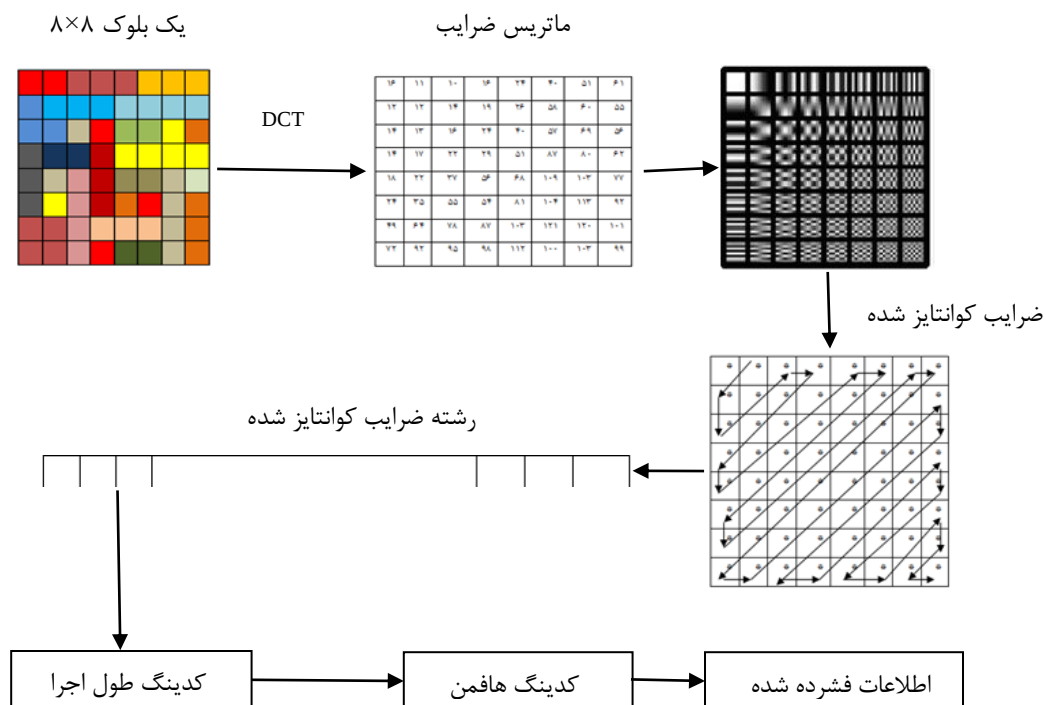
در نوع با اتلاف، تصویر به بخش‌هایی تقسیم شده و از ایده‌ی حذف تفاوت نامحسوس رنگ‌های نزدیک به هم استفاده می‌شود [۲۸]. همانطور که در شکل (۲-۱۶) نشان داده شده است، تفاوت نامحسوس رنگ‌ها در یک بخش از تصویر، توسط این الگوریتم از بین می‌رود.



شکل ۲-۱۶: حذف تفاوت نامحسوس رنگ‌های نزدیک به هم در تکنیک فشرده‌سازی JPEG.

در این فرمت ذخیره‌سازی، میزان فشرده‌سازی و میزان اطلاعاتی که حذف می‌شوند، قابل تعیین خواهد بود. هر چه تصویر بیشتر متراکم شود، کیفیت تصویر پایین‌تر است. به طور خلاصه، در این روش مراحل زیر انجام می‌شود [۲۹]:

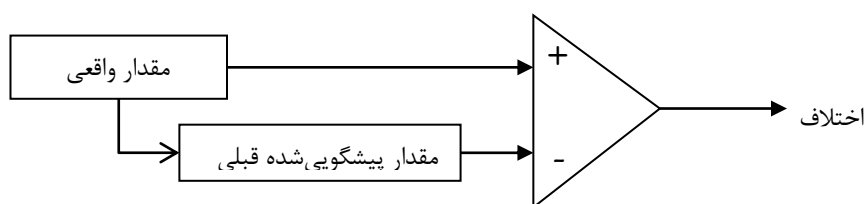
ابتدا تصویر به بلوک‌هایی با ابعاد 8×8 تقسیم شده و تبدیل گسسته کسینوسی به‌طور جداگانه به تمام بلوک‌ها اعمال می‌شود. با استفاده از جدول کوانتایز، ضرایب حاصل از تبدیل کسینوسی، کوانتایز می‌شوند. در این مرحله، قسمتی از اطلاعات از بین می‌رود. جدول کوانتایز بر اساس سیستم بینایی انسان ایجاد شده است. در این جدول سعی شده که مولفه‌هایی که از نظر سیستم بینایی انسان ارزش‌ترند بیشتر حفظ شوند. بنابراین ضرایب متناظر با مولفه‌های مهم‌تر، کوچک‌تر هستند [۳۰]. پس از کوانتایز کردن، ضرایب با استفاده از الگوی زیگزاک مرتب شده و در مراحل بعد با استفاده از کدینگ طول اجرا و کدینگ هافمن، هرچه بیشتر فشرده و ذخیره می‌گردند. این مراحل، در شکل (۲-۱۷) نشان داده شده است.



شکل ۲-۱۷: مراحل روش JPEG.

۲-۴-۲- فرمت ذخیره‌سازی Lossless JPEG

در نوع با اتلاف JPEG، به دلیل استفاده از تبدیل گسسته‌ی کسینوسی، بخشی از اطلاعات از بین می‌رود. زیرا در عکس این تبدیل، داده‌ها به طور کامل بازیابی نمی‌شوند. در Lossless JPEG، به جای استفاده از تبدیل گسسته‌ی کسینوسی، از کدینگ پیشگویانه (DPCM)^۱ استفاده می‌شود. مطابق شکل (۲-۱۸)، در این روش، رمزکننده مقدار پیشگویی شده را با مقدار واقعی پیکسل در محل X مقایسه می‌کند و سپس اختلاف موجود را با استفاده از یکی از روش‌های بدون اتلاف رمز می‌کند [۳۱]. برای پیشگویی مقدار هر پیکسل، از ترکیبی از مقادیر حداکثر تا سه همسایه آن پیکسل استفاده می‌شود. شکل (۲-۱۹) موقعیت همسایه‌های پیکسل X، را نشان می‌دهد. جدول (۲-۱) نیز هشت الگوی قابل استفاده برای پیشگویی را معرفی می‌کند [۳۲].



شکل ۲-۱۸: مقایسه مقدار پیشگویی شده با مقدار واقعی پیکسل در روش DPCM.

	C	B	D
	A	X	

شکل ۲-۱۹: موقعیت همسایه‌های پیکسل X.

^۱ Differential Pulse-Code Modulation

جدول ۱-۲: هشت الگوی پیشگویی در روش Lossless JPEG

مقدار	نوع پیشگویی
0	بدون پیشگویی
1	A
2	B
3	C
4	A+B-C
5	$A+(B-C)/2$
6	$B+(A-C)/2$
7	$(A+B)/2$

۲-۴-۳- فرمت ذخیره سازی JPEG_LS

این الگوریتم برای اولین بار توسط Weinberger در سال ۲۰۰۰ ارائه شد. این روش از پیچیدگی پایین برخوردار بوده و می تواند به صورت بدون اتلاف و نزدیک به بی اتلاف مورد استفاده قرار گیرد. در حالت دوم، یک محدوده خطای پیش بینی در فاز پیش بینی در نظر گرفته می شود. این الگوریتم از دو بخش مجزای Modeler و Coder تشکیل شده است. برای فشرده کردن یک پیکسل مانند x ، این پیکسل و پیکسل های همسایه اش که در شکل (۲-۲۰) نشان داده شده اند، وارد بخش modeler می شوند.

c	b	d
a	x	

شکل ۲-۲۰: موقعیت پیکسل X و پیکسل های همسایه

در شکل (۲-۲۱) در بخش Gradient پارامترهای G_1 ، G_2 و G_3 محاسبه می شوند. نحوه ی محاسبه -

ی این مقادیر در فرمول (۲-۶) تا (۲-۸) آمده است.

$$G_1 = d - b \quad (۲-۶)$$

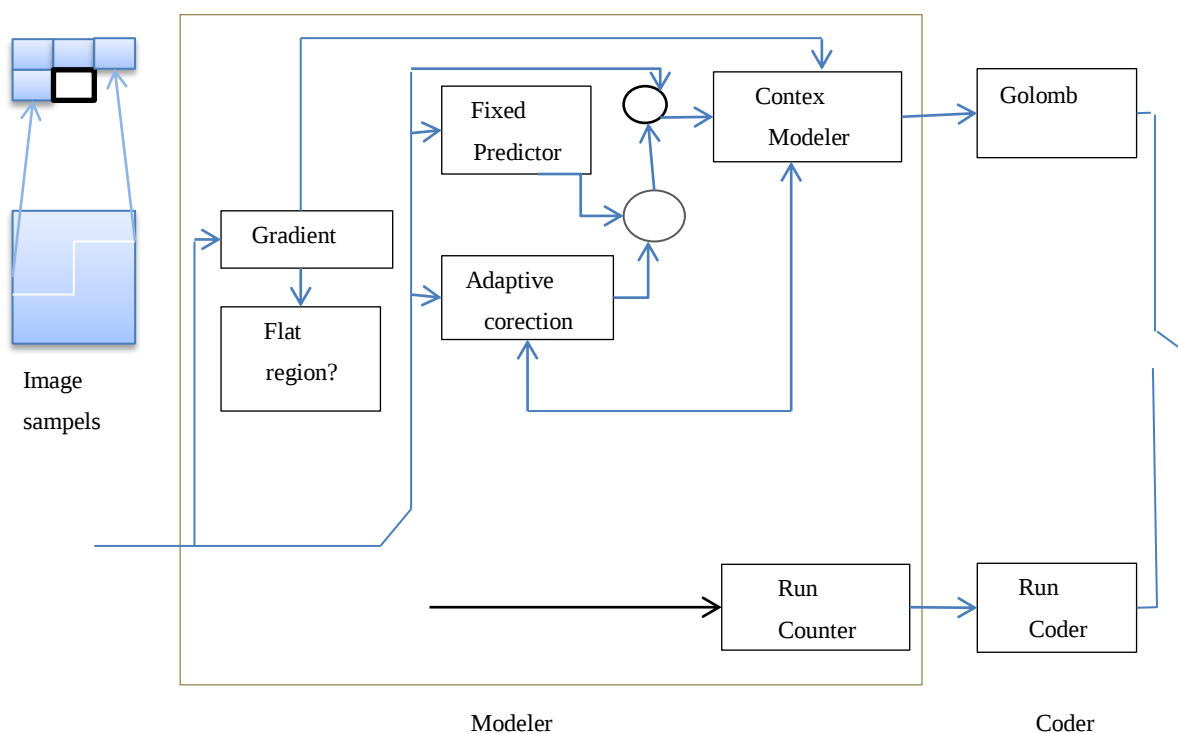
$$G_2 = b - c \quad (۲-۷)$$

$$G_3 = c - a \quad (۲-۸)$$

اگر $G1=G2=G3=0$ برقرار باشد الگوریتم وارد مرحله‌ی run شده و تا زمانی که x برابر a است و تا انتهای ردیف تصویر، تعداد این تساوی‌ها شمارش می‌شود. در غیر این صورت الگوریتم وارد بخش regular شده و در بلوک Fixed prediction با استفاده از فرمول (۹-۲) مقدار x پیش‌بینی می‌شود.

$$X_{pr} = \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b) \\ \max(a, b) & \text{if } c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases} \quad (9-2)$$

در Context modeler مقادیر کوانتایز شده و برداری برای رخداد مورد نظر به دست می‌آید. این بردار هر Context را آدرس‌دهی می‌کند تا امکان بهبود خطای پیش‌بینی ثابت توسط بخش Adaptive Correction فراهم شود. برای از بین بردن مقادیر منفی خطا، تمام مقادیر به مقادیر مثبت نگاشت یافته و به بخش Coder ارسال می‌گردند. در بخش Coder این مقادیر توسط کدکننده‌ی گولمب^۱ به کد با طول متغیر تبدیل می‌شوند [۳۳-۳۴].



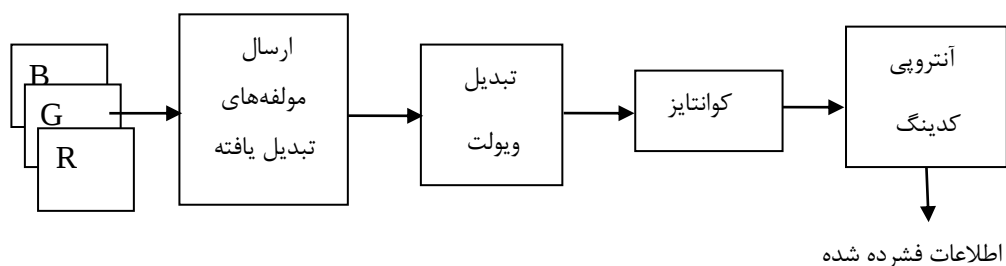
شکل ۲-۲۱: دیاگرام الگوریتم JPEG_LS

¹ Golomb

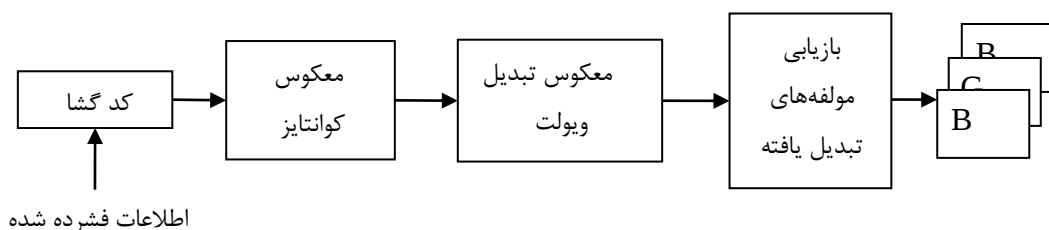
۲-۴-۴- فرمت ذخیره سازی JPEG2000

این استاندارد در سال ۲۰۰۱ نهایی شد. در این تکنیک یک روش کدگذاری مبتنی بر ویولت^۱ استفاده شده است. در JPEG2000 در ابتدا تصویر به بلوک‌هایی تقسیم شده و هر بلوک به طور مجزا به مجموعه‌ای از فیلترهای ویولت، ارسال و به ضرایب ویولت تبدیل می‌شود. در نتیجه باندهای فرعی ایجاد می‌گردد که هر باند فرعی، دارای دامنه فرکانس عمودی و افقی از تصویر است. اطلاعات با جزئیات کمتر و فرکانس پایین‌تر در مرحله‌ی اول و اطلاعات با فرکانس‌های بالاتر، در مراحل بعدی گنجانده می‌شوند [۳۶-۳۵]. در مرحله‌ی بعد ضرایب ویولت کوانتایز می‌شوند. سپس مقادیر با کدینگ آنتروپی کد شده و در نهایت تصویر فشرده شده، آماده‌ی ارسال یا ذخیره خواهد بود. این روند در شکل (۲-۲۲) نشان داده شده است. مراحل بازبینی درست عکس مراحل فشرده‌سازی می‌باشد (شکل (۲-۲۳)). از جمله مزیت‌های این فرمت ذخیره‌سازی عبارتند از [۳۸-۳۷]:

- ✓ انجام فشرده‌سازی با حداقل افت کیفیت
- ✓ انجام فشرده‌سازی با نرخ فشرده‌سازی بالا
- ✓ امکان استفاده به صورت با اتلاف و بدون اتلاف



شکل ۲-۲۲: مراحل فشرده‌سازی در الگوریتم JPEG2000



شکل ۲-۲۳: مراحل بازبینی در الگوریتم JPEG2000

¹ Wavelet

۲-۴-۵- فرمت ذخیره‌سازی GIF

فشرده‌سازی^۱ GIF یک فرمت فشرده‌سازی بدون‌اتلاف است. از این فرمت برای ذخیره‌سازی فایل‌های ترسیمی، متن و حتی تصاویر کوچک استفاده می‌شود. برای مشخص شدن نحوه‌ی عملکرد این فرمت، تصویر ۶×۶ شکل (۲-۲۴) که از سه رنگ a، b و c ساخته شده است را در نظر بگیرید.

c	c	c	c	c	c
c	b	b	b	b	b
c	b	b	b	b	b
c	a	a	b	b	b
c	a	a	b	b	b
c	c	c	c	c	c

شکل ۲-۲۴: تصویری متشکل از سه رنگ.

یک راه توصیف این تصویر به این صورت است:

سطر ۱: cccccc

سطر ۲: cbbbbbb

سطر ۳: cbbbbbb

این روند توصیف به همین صورت تا سطر آخر ادامه می‌یابد. در این روش هر پیکسل یک بار و در

یک زمان تعریف می‌شود. برای توصیف بهتر می‌توان به‌صورت زیر نیز عمل نمود.

سطر ۱: 6c

سطر ۲: c5b

سطر ۳: c5b

این فرمت ذخیره‌سازی، زمانی که پیکسل‌ها متنوع باشند، نرخ فشرده‌سازی بالایی را حاصل نمی‌کند. به‌طور کلی ایده‌ی فشرده‌سازی GIF حول این محور ساده است که تعداد رنگ‌ها یا تعداد

تغییرات رنگ در فایل کاهش پیدا کند. بنابراین، این فرمت برای تصاویری که تعداد رنگ کمتر از ۲۵۶

¹ Graphics Interchange Format

دارند و همچنین تصاویر سطح خاکستری مناسب است. از این فرمت ذخیره‌سازی برای ذخیره‌ی تصاویر ساده مثل لوگو، تصاویر انیمیشن و کارتونی استفاده می‌شود.

۲-۵- ویژگی‌های یک فضای رنگ مناسب در فشرده‌سازی تصویر

یک تصویر شامل مقادیری از پیکسل‌ها با شدت روشنایی متفاوت است. تصاویر در بازه‌ی (۰-۲۵۵) برای هر یک از مولفه‌های قرمز، سبز و آبی تشکیل دهنده‌ی تصاویر رنگی و تصاویری در بازه‌ی (۰-۲۵۵) تصاویر خاکستری و تصاویری با مقادیر صفر و یک، تصاویر باینری را ایجاد می‌کنند. برای نمایش یک تصویر رنگی، فضاهای مختلفی ایجاد و معرفی شده‌اند که هر یک کاربرد خاص خود را دارند و بسته به نوع کاربرد، مورد استفاده قرار می‌گیرند. در فشرده‌سازی تصویر نیز سعی بر این است که از فضای رنگی استفاده شود که دارای ویژگی‌های زیر باشد:

✓ اکثر انرژی در یک کانال تجمع داشته باشد. (با وجود این ویژگی، می‌توان دو کانال دیگر را به میزان بیشتری فشرده کرد و کانالی را که اطلاعات بیشتری در بردارد کمتر فشرده نمود).

✓ میزان وابستگی^۱ درون پیکسل‌های هر کانال بیشتر باشد.

✓ میزان وابستگی بین کانال‌های مختلف کم باشد.

۲-۶- نتیجه‌گیری

امروزه تکنیک‌های فشرده‌سازی زیادی ایجاد شده‌اند که سعی دارند داده‌ها را به بهترین شکل ممکن فشرده کنند. برخی از این تکنیک‌ها برای رسیدن به نرخ فشرده‌سازی بیشتر، بخشی از اطلاعات را از بین می‌برند و برخی دیگر با نرخ فشرده‌سازی کمتر، تمام اطلاعات تصویر را حفظ می‌کنند. در این فصل مروری بر برخی تکنیک‌های رایج فشرده‌سازی تصویر داشتیم. همچنین نحوه‌ی عملکرد چند فرمت ذخیره‌سازی تصویر را شرح دادیم. با توجه به مطالب این فصل، می‌توان دریافت که هر الگوریتم

¹ Correlation

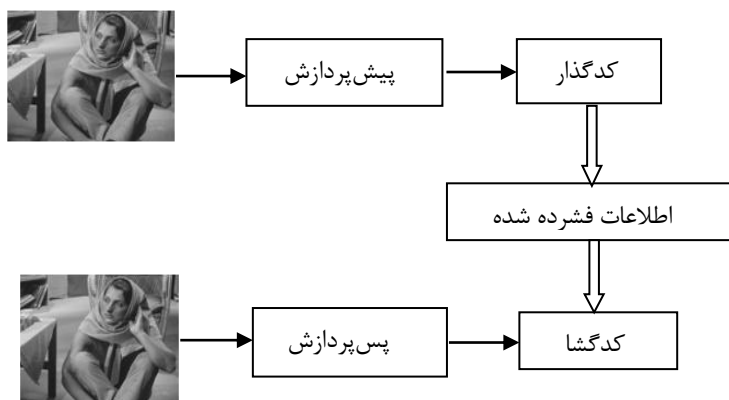
فشرده‌سازی تکنیک منحصر به فرد خود را دارد، اما در عین حال هدف همه‌ی این روش‌ها، شناسایی افزونگی تصویر براساس یک یا چند ویژگی و در نتیجه فشرده‌سازی آن است.

فصل سوم

مروری بر روش‌های پیشین پیش‌پردازش

۳-۱- مقدمه

به دلایلی که در فصل اول مطرح شد، انجام یک پیش‌پردازش مناسب بر روی تصویر، می‌تواند در بهبود نرخ فشرده‌سازی موثر باشد. در واقع با انجام پیش‌پردازش، تصویر برای مرحله‌ی فشرده‌سازی آماده خواهد شد. همانطور که در شکل (۳-۱) نشان داده شده است، فاز پیش‌پردازش، قبل از انجام فشرده‌سازی صورت می‌گیرد. در بازیابی تصویر، پس از آنکه تصویر از حالت فشرده خارج شد، عکس عمل پیش‌پردازش بر روی آن اعمال می‌شود.



شکل ۳-۱: دیاگرام مربوط به فشرده‌سازی و بازیابی تصویر به همراه فاز پیش‌پردازش.

تاکنون روش‌هایی در زمینه‌ی پیش‌پردازش تصویر به منظور افزایش نرخ فشرده‌سازی، ارائه شده است. در این فصل، مروری بر برخی از پژوهش‌های موجود خواهیم داشت.

۳-۲- مروری بر پژوهش‌های موجود

در [۳۹]، یک روش با اتلاف، با استفاده از فیلتر گوسین به منظور بهبود عملکرد JPEG مطرح شده است. با استفاده از این فیلتر، تصویر اولیه به یک تصویر هموار^۱ تبدیل می‌شود. تصاویر هموار، می‌توانند با نرخ بالاتری نسبت به تصاویر با جزئیات فرکانس بالا فشرده شوند. در واقع جزئیات موجود در تصویر باعث افزایش مقادیر فرکانس بالا در تبدیلاتی نظیر DCT یا DWT می‌گردند، که این امر منجر

^۱ Smooth

به افزایش تعداد ضرایب معنادار در حوزه‌ی تبدیل می‌شود. از آنجا که این ضرایب برای فشرده‌سازی باید کد شوند، هر چه تعدادشان بیشتر باشد، میزان فشرده‌سازی کمتر خواهد بود. لذا در این روش، با هموار کردن تصویر سعی شده تا جزئیات و تعداد ضرایب معنادار کاهش یابد.

اگرچه حذف جزئیات باعث افزایش میزان فشرده‌سازی خواهد شد اما از طرف دیگر کیفیت تصویر را کاهش خواهد داد. برای اینکه بتوان تصویر را بازیابی نمود، لازم است که عکس عمل این فیلتر بر روی آن اعمال شود، اما چون فیلتر گوسین قابلیت برگشت‌پذیری بالایی ندارد، تصویر بازیابی شده با تصویر اولیه اندکی تفاوت دارد. لذا یک فاز پس‌پردازش^۱ نیز برای بهبود کیفیت تصویر در نظر گرفته شده است. با انجام این مرحله نیز، میزان خرابی تصویر، زیاد خواهد بود، به همین دلیل در این روش میزان مات کردن تصویر توسط فیلتر گوسین باید اندک باشد. اما مات کردن به میزان اندک هم سبب بهبود چندانی در نرخ فشرده‌سازی نمی‌شود.

در [۴۰] روش پیش‌پردازش بدون اتلافی بر مبنای متراکم کردن هیستوگرام^۲ تصویر ارائه شده است. متراکم کردن هیستوگرام از جمله پیش‌پردازش‌هایی است که می‌تواند منجر به بهبود نرخ فشرده‌سازی شود [۴۱]. با این پیش‌پردازش، تمامی سطوح خاکستری تصویر به یک محدوده مشخص مجاور هم انتقال می‌یابند، به‌طوری‌که فاصله هر سطح خاکستری با نزدیکترین سطح خاکستری موجود در تصویر، یک است. در واقع این پیش‌پردازش سبب می‌شود که تغییرات پیکسل‌های مجاور تصویر کم گردد. برای بازیابی تصویر، نیاز است که جدول نگاشت مربوط به متراکم کردن هیستوگرام تصویر را ذخیره نمود. این تصویر پیش‌پردازش شده، را به یک روش بی‌اتلاف مانند JPEG-LS و یا JPEG2000 بدون اتلاف اعمال نموده تا تصویر فشرده گردد. این روش توانسته حدود ۲۷/۳ درصد نرخ فشرده‌سازی JPEG2000 بدون اتلاف را افزایش دهد. قابل ذکر است که این تکنیک، فقط برای

^۱ Post Processing

^۲ Packing histogram

تصاویری با هیستوگرام خلوت^۱ کاربرد دارد. همچنین وقتی همه‌ی شدت روشنایی‌ها، در تصویر وجود داشته باشند، متراکم کردن هیستوگرام به صورت کلی نمی‌تواند منجر به بهبود نرخ فشرده‌سازی شود. به همین دلیل در [۴۲] یک روش پیش‌پردازش بدون اتلاف بر مبنای متراکم کردن محلی هیستوگرام، ارائه شده است. در این روش، تصویر به ابعاد ثابت 32×32 تقسیم شده و متراکم کردن هیستوگرام برای هر یک از بخش‌های تصویر انجام می‌شود. در یک بخش کوچک از تصویر، احتمال حضور تمام شدت روشنایی‌ها اندک است، لذا با متراکم کردن هیستوگرام به صورت محلی می‌توان نرخ فشرده‌سازی را افزایش داد. در این روش، لیستی از شدت روشنایی‌های مورد استفاده در هر بلوک ثبت می‌شود تا در مرحله‌ی بازیابی مورد استفاده قرار گیرد. اشکال این روش این است که سایز بلوک‌ها ثابت در نظر گرفته شده و در نتیجه تعداد بلوک‌ها زیاد است. از آنجا که برای هر بلوک یک جدول نگاشت در نظر گرفته می‌شود، حجم این جداول، سربار زیادی ایجاد می‌کند. لذا این روش، بهبود قابل توجهی در نرخ فشرده‌سازی نخواهد داد. تصویر پردازش شده باید با یک روش بدون اتلاف فشرده شود. روش فشرده‌سازی مورد استفاده در این مرجع JPEG-LS است که پیش‌پردازش انجام شده منجر به بهبود ۱۵ درصدی این روش شده است.

نسخه‌ی بهبود یافته‌ی مرجع [۴۲] در [۴۳] ارائه شده است. در این روش ابعاد بلوک‌ها متغیر است. در واقع بلوک‌بندی تصویر با یک سایز ثابت، ممکن است بهینه نباشد. به عبارت دیگر، برای نواحی یکنواخت، ابعاد بلوک بزرگتر و برای نواحی با بافت متراکم، ابعاد بلوک کوچکتر در نظر گرفته می‌شود. در این روش، شکل بلوک‌ها نیز ثابت نخواهد بود، بنابراین برای بازیابی، تصویری که در بردارنده‌ی شکل و مرزهای هر بلوک می‌باشد نیز ذخیره می‌گردد. مثالی از عملکرد این روش به همراه تصویر لازم برای بازیابی، در شکل (۳-۲) نشان داده شده است. روش فشرده‌سازی مورد استفاده در این مرجع، JPEG-LS است که پیش‌پردازش انجام شده، منجر به بهبود ۱۷ درصدی این الگوریتم شده است.

¹ Sparse histogram



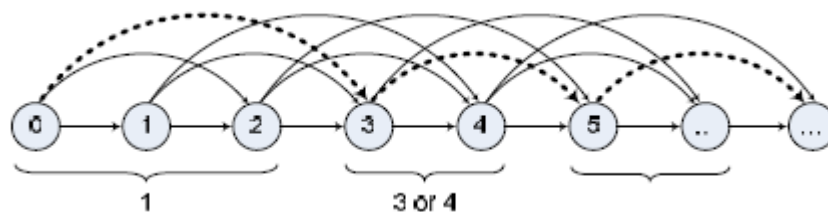
شکل ۳-۲: مثالی از عملکرد روش پیشنهاد شده در مرجع [۴۳]. (الف) تصویر اصلی، (ب)، تصویر پردازش شده، (ج) تصویر ذخیره شده برای بازیابی تصویر.

در [۴۴]، یک روش شبه‌بی‌اتلاف [۴۵] برای بهبود JPEG-LS مطرح شده است. در این روش یک گراف ایجاد شده و برای هر پیکسل و همسایه‌های آن، تعدادی گره در این گراف در نظر گرفته می‌شود. گره‌ها مقادیری برابر یا نزدیک به شدت روشنایی آن پیکسل دارند. برای هر سطر از تصویر یک شبکه از این مقادیر همسایه ایجاد می‌شود. خطای پیشگویی بین هر دو گره، محاسبه شده و به عنوان برچسب هر کمان در نظر گرفته می‌شود. در این روش یک الگوریتم حریصانه نیز مطرح شده و برای انتخاب مسیری با بیشترین تعداد خطای پیشگویی یکسان، از این الگوریتم حریصانه استفاده شده است. سپس دنباله‌ی به‌دست آمده با استفاده از کدینگ طول اجرا فشرده می‌شود. این روش منجر به بهبود ۱۴/۲۲ درصدی روش JPEG-LS شده است. اما پیچیدگی و زمان پردازش این روش بالاست.

در [۴۶] روش شبه‌بی‌اتلافی معرفی می‌شود که بر مبنای متراکم‌سازی محلی هیستوگرام است. در هیستوگرام تصویر، هر سطح خاکستری به همراه نمودار فراوانی مربوط به آن، یک "بین"^۱ نامیده می‌شود. در این روش ابتدا سعی شده تا تعداد "بین" موجود در هیستوگرام کاهش یابد. یعنی تعدادی از آنها به "بین"های مجاورشان انتقال پیدا کنند. برای دستیابی به مناسب‌ترین حالت انتقال "بین"ها، از گراف استفاده می‌شود. هر شدت روشنایی با یک گره در گراف نمایش داده شده و مجموعه شدت روشنایی‌هایی که باید در یک دسته قرار بگیرند نیز با یک کمان در گراف مشخص می‌شوند.

شکل (۳-۳) مثالی از گراف تعریف شده را نشان می‌دهد. در این شکل هر کمان مشخص می‌کند که انتقال، به چه صورت انجام گیرد.

¹Bin



شکل ۳-۳: نمونه‌ای از گراف تعریف شده در مرجع [۴۶].

پس از ایجاد گراف، از الگوریتم ویتربی^۱ برای یافتن مسیری با مینیمم آنتروپی استفاده می‌شود [۴۷]. بر اساس این مسیر، تصویر، پردازش شده و هیستوگرامی با تعداد "بین" کمتر ایجاد می‌شود. این عمل، سبب کاهش آنتروپی شده و وابستگی بین پیکسل‌های تصویر را افزایش می‌دهد. در نتیجه، تصویر پردازش شده با نرخ بیشتری فشرده می‌شود. این روش منجر به بهبود عملکرد JPEG-LS شده است. همچنین نتایج مرجع [۴۴] را تا حدی بهبود داده است.

این پیش‌پردازش، دارای پیچیدگی و زمان پردازش بالاست. فرض کنیم، هر "بین" می‌تواند تنها به "بین" قبل یا بعد از خودش انتقال یابد و یا موقعیت آن تغییر نکند. با این فرض، امکان جایگزینی مقدار هر پیکسل با پیکسل قبل یا بعد از خودش وجود دارد و یا اینکه پیکسل مقدار قبلی خودش را حفظ می‌کند، در این شرایط سه حالت انتخاب وجود خواهد داشت. اگر هر پیکسل، با هشت بیت نمایش داده شود ۲۵۶ سطح خاکستری نیز خواهیم داشت و حالات مختلف برای تغییر پیکسل‌ها در تصویر برابر 3^{256} خواهد بود. از طرفی تشکیل چنین گرافی و پیدا کردن مسیر بهینه بخش دیگری است که نیازمند صرف زمان است.

در [۴۸] روش پیش‌پردازش بدون اتلافی با استفاده از ماتریس هم‌رخداد^۲ تصویر، به منظور کاهش تنوع مقادیر پیکسل‌ها انجام شده است. ماتریس هم‌رخداد، ماتریسی است که دربردارنده‌ی اطلاعات بافتی یک تصویر می‌باشد. مزیت استفاده از این ماتریس برای نمایش بافت تصویر، این است که، موقعیت مکانی پیکسل‌ها نیز در نظر گرفته می‌شود. عناصر ماتریس هم‌رخداد G_{ij} ، نشان‌دهنده‌ی

^۱ Viterbi Algorithm

^۲ Occurrence Matrix

فرکانس نسبی دو پیکسل، یکی با سطح خاکستری i و دیگری با سطح خاکستری z در فاصله‌ی خاص و در یک جهت تعیین شده در تصویر می‌باشند [۴۹].

در این روش، ماتریسی ایجاد می‌شود که نسبت به ماتریس تصویر اولیه، دارای افزونگی بیشتری است. در نتیجه قابلیت فشرده شدن با نرخ بیشتری را داراست. روش مطرح شده قادر است نرخ فشرده‌سازی هافمن و کدینگ حسابی را به میزان $37/85$ درصد افزایش دهد. قابل ذکر است که این نتایج با در نظر گرفتن سربار^۱ ماتریس هم‌رخداد بیان شده و این سربار همواره بهینه نخواهد بود و به ابعاد تصویر بستگی دارد. یعنی ممکن است در برخی موارد سربار حاصل از ماتریس هم‌رخداد، بسیار بیشتر از میزان کاهش حجمی باشد که از پیش‌پردازش مطرح شده، حاصل می‌شود. این امر بخصوص در پردازش تصاویر با ابعاد کوچکتر از 256×256 رخ می‌دهد.

در [۵۰] روش پیش‌پردازش بدون اتلافی ارائه شده است که منجر به افزایش نرخ فشرده‌سازی الگوریتم‌های بدون اتلاف برای تصاویر رادیوگرافی صنعتی شده است. در این پژوهش، تصویر در چهار حالت مختلف، به عنوان ورودی الگوریتم فشرده‌سازی، در نظر گرفته می‌شود. تصویر باینری، تصویر سطح خاکستری، صفحات تک بیتی تصویر در حالت باینری و صفحات تک بیتی تصویر در حالت کدگذاری گری^۲، به طور جداگانه به عنوان ورودی در نظر گرفته می‌شوند. پس از دریافت تصویر، هشت نوع پیمایش بر روی آن اعمال می‌گردد. در نهایت با استفاده از معیارهای میانگین طول اجرا^۳ [۵۱]، معیار فعالیت^۴ [۵۱] و معیار مجموع تفاوت^۵ [۵۱]، بهترین نوع پیمایش انتخاب می‌شود. بر اساس بهترین پیمایش انتخابی، تصویر جدیدی ساخته شده و در نهایت تصویر پردازش شده با کدینگ طول اجرا و سپس هافمن فشرده می‌شود. در نهایت بیان شده است که پیمایش مارپیچی^۶ برای تصویری که

¹ Overhead

² Gray Code

³ Average Run Length

⁴ Activity Measure

⁵ Sum of Difference

⁶ Spiral

صفحات تک بیتی آن به صورت گری کد شده است، نسبت به سایر موارد، منجر به نرخ فشرده‌سازی بیشتری خواهد شد. این پژوهش بر روی یک نوع خاص از تصاویر رادیوگرافی صنعتی انجام شده است و ممکن است نتایج بیان شده قابل تعمیم بر روی تصاویر دیگر نباشد.

در [۵۲]، روش‌های پیش‌پردازش بدون اتلافی برای بهبود فشرده‌سازی تصاویر پزشکی ارائه می‌شود. پایه‌ی این روش بر تبدیلی با نام باروس و ویلر^۱ استوار است [۵۳]. روش باروس و ویلر اولین بار برای فشرده‌سازی متن مورد استفاده قرار گرفت. این تکنیک به‌تنهایی حجم را کاهش نمی‌دهد بلکه نحوه‌ی چیدمان سمبل‌ها را به‌گونه‌ای تغییر می‌دهد که تعداد دنباله‌ها با سمبل یکسان بیشتری ایجاد شده و در نتیجه فشرده‌سازی با نرخ بیشتری انجام شود [۵۴-۵۵]. در این مرجع متدهای ترکیبی مختلفی بر پایه‌ی تبدیل معرفی شده ارائه می‌شود که می‌تواند در برخی موارد نتایج فشرده‌سازی الگوریتم‌های JPEG2000 و JPEG_LS را برای تصاویر پزشکی بهبود بخشد.

۳-۳- نتیجه‌گیری

در این فصل روش‌های پیش‌پردازش موجود مرور شد. اگرچه در زمینه‌ی روش‌های فشرده‌سازی و ارائه‌ی الگوریتم‌های فشرده‌سازی جدید پژوهش‌های زیادی انجام شده است، اما روش‌هایی که به‌طور خاص تنها بر مساله‌ی پیش‌پردازش تصویر، تاکید داشته باشند اندک است. در پیش‌پردازش تصویر به منظور بهبود نرخ فشرده‌سازی باید پیش‌پردازش تا حد امکان دارای ویژگی‌های زیر باشد:

- ✓ قابلیت بازیابی تصویر اولیه وجود داشته باشد.
- ✓ زمان پردازش اندک باشد
- ✓ پیچیدگی روش پایین باشد.
- ✓ در اکثر موارد برای بازیابی تصویر، باید اطلاعاتی ذخیره شوند. حفظ این اطلاعات سرباری ایجاد می‌کند. پس باید همواره در نظر داشت که سربار ایجاد شده، در قبال نرخ بهبود، قابل چشم‌پوشی باشد. در غیر این صورت پردازش مفید نخواهد بود.

¹ Burrows Wheeler transform

✓ روش انجام شده تا حد امکان قابلیت استفاده برای انواع تصاویر را داشته باشد.

روش‌های مطرح شده در این فصل عمدتاً، به یک یا دو مورد از نکات ذکر شده توجه نموده‌اند، در- حالیکه در این پایان‌نامه تلاش نمودیم به اکثر این موارد توجه نموده و با در نظر گرفتن آنها پیش- پردازش‌هایی مناسب‌تر را بر روی تصاویر انجام دهیم. این روش‌ها در فصل بعد معرفی خواهند شد. در جدول (۱-۳)، خلاصه‌ای از عملکرد روش‌های موجود، آورده شده است.

جدول ۱-۳: خلاصه‌ای از عملکرد روش‌های پیش‌پردازش موجود.

مرجع	نوع روش	زمان پردازش	پیچیدگی	قابلیت اجرا برای انواع تصویر	درصد بهبود	الگوریتم بهبود یافته
[۳۹]	با اتلاف	کم	کم	دارد	-	-
[۴۰]	بدون اتلاف	کم	کم	ندارد	۲۷/۳	JPEG_2000
[۴۲]	بدون اتلاف	متوسط	متوسط	دارد	۱۵	JPEG_LS
[۴۳]	بدون اتلاف	متوسط	متوسط	دارد	۱۷	JPEG_LS
[۴۴]	شبه بی‌اتلاف	بالا	بالا	دارد	۱۴/۲۲	JPEG_LS
[۴۶]	شبه بی‌اتلاف	بالا	بالا	دارد	-	JPEG_LS
[۴۸]	بدون اتلاف	کم	متوسط	ندارد	۳۷/۸۵	هافمن و کدینگ حسابی
[۵۰]	بدون اتلاف	متوسط	متوسط	ندارد		هافمن

فصل چهارم

معرفی و ارزیابی روش‌های پیشنهادی

۴-۱- مقدمه

همانطور که در فصل اول گفته شد، پیش‌پردازش تصویر می‌تواند نقش مهمی در بهبود عملکرد الگوریتم‌های فشرده‌سازی داشته باشد. در این بخش به معرفی و ارزیابی روش‌های پیش‌پردازش پیشنهادی خواهیم پرداخت. در مجموع چهار روش معرفی شده‌اند. در این روش‌ها سعی شده پیش‌پردازش‌ها، بدون اتلاف، یا تا حد امکان نزدیک به بی‌اتلاف باشند. این پیش‌پردازش‌ها توانسته‌اند تا حد قابل قبولی عملکرد الگوریتم‌های JPEG، JPEG2000 و Lossless JPEG که از پرکاربردترین و توانمندترین الگوریتم‌های فشرده‌سازی می‌باشند را بهبود دهند.

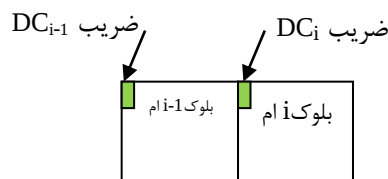
۴-۲- نحوه‌ی بکارگیری ضرایب DC در JPEG

از آنجا که روش پیشنهادی اول بر اساس نحوه‌ی عملکرد الگوریتم JPEG ارائه شده است. در این بخش به توضیح نحوه‌ی بکارگیری ضرایب DC در الگوریتم JPEG خواهیم پرداخت. پس از اعمال تبدیل گسسته کسینوسی دو بعدی و کوانتایز کردن، ضرایب با مقادیر صفر بسیاری در یک بلوک ایجاد می‌شوند. به این مقادیر، ضرایب AC گفته می‌شود. یک بلوک از تصویر پس از اعمال تبدیل گسسته‌ی کسینوسی و کوانتایز کردن در شکل (۴-۱) نشان داده شده است. اولین ضریب در گوشه‌ی سمت چپ و بالا، همان مقدار ضریب DC مربوط به هر بلوک است. الگوریتم فشرده‌سازی JPEG کدینگ تفاضلی^۱ را بر روی ضرایب DC اعمال می‌کند. کدینگ تفاضلی ضرایب DC به صورت تفاضل ضریب DC بلوک i ام از ضریب DC بلوک $i-1$ ام بیان می‌شود (شکل (۴-۲)).

¹ Differential Coding

DC	۱۰	-۶	۲	۰	۰	-۱	۰
۴	-۵	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰

شکل ۴-۱: مقادیر یک بلوک از تصویر پس از اعمال تبدیل گسسته‌ی کسینوسی و کوانتایز کردن ضرایب.



شکل ۴-۲: ضرایب DC دو بلوک مجاور

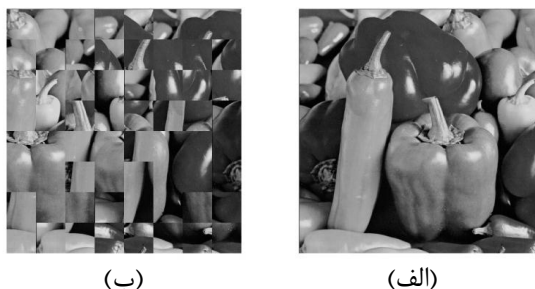
پیش‌پردازش پیشنهادی اول هم بر اساس نحوه‌ی بکارگیری ضرایب DC ارائه شده است. در ادامه به توضیح این روش می‌پردازیم.

۴-۳- جابجایی نواحی تصویر بر اساس ضرایب DC، به منظور

کاهش دامنه‌ی تغییرات (روش پیشنهادی اول)

همانطور که در بخش (۴-۲) بیان شد، در الگوریتم فشرده‌سازی JPEG تفاضل ضرایب DC هر بلوک از ضرایب DC بلوک مجاورش به دست می‌آید. بنابراین تصویر را به گونه‌ای پردازش می‌کنیم که مقادیر این تفاضل کاهش یابد. روش پیشنهادی یک روش ساده، بدون اتلاف و با پیچیدگی پایین است. پردازش در دو گام انجام می‌شود. در گام اول، تصویر به بلوک‌های بدون هم‌پوشانی تقسیم می‌شود. سپس ضرایب DC هر یک از بلوک‌ها محاسبه شده و تمام بلوک‌هایی که در یک ردیف قرار دارند بر اساس افزایش (کاهش) مقدار ضرایب DC مرتب می‌شوند.

شکل (۴-۳) یک تصویر بلوک‌بندی شده با ابعاد 64×64 را نشان می‌دهد که در آن بلوک‌های هر ردیف بر اساس ضرایب DC مرتب شده‌اند.



شکل ۳-۴: بلوک‌بندی و جابجایی بلوک‌ها بر اساس ضرایب DC. (الف) تصویر اصلی، (ب) تصویر پردازش شده.

از آنجا که در روش JPEG، بلوک‌بندی تصویر با ابعاد 8×8 صورت می‌گیرد. در این پردازش نیز تصویر به بلوک‌هایی با همین ابعاد تقسیم می‌شود. بر همین اساس برای یک تصویر با ابعاد $m \times n$ تعداد $m/8$ بلوک در جهت افقی و $n/8$ بلوک در جهت عمودی ایجاد می‌گردد. بلوک‌های هر ردیف به طور مجزا مرتب می‌شوند، بنابراین برای بازیابی کافی است به هر بلوک قبل از جابجایی، عددی بین $[1-m/8]$ اختصاص دهیم که بیانگر محل بلوک در جهت افقی است و باید شماره‌ی آنها حفظ شود. لذا میزانی سربار خواهیم داشت. برای به حداقل رساندن میزان سربار، مقدیری که بیانگر شماره‌ی بلوک‌ها هستند در یک ماتریس ذخیره نموده و سپس ماتریس ذکر شده را در قالب یک فایل تصویری با فرمت PNG ذخیره می‌کنیم. پس از پردازش تصویر، هر بلوک با استفاده از این شماره اختصاص داده شده، به محل قبلی خود باز می‌گردد.

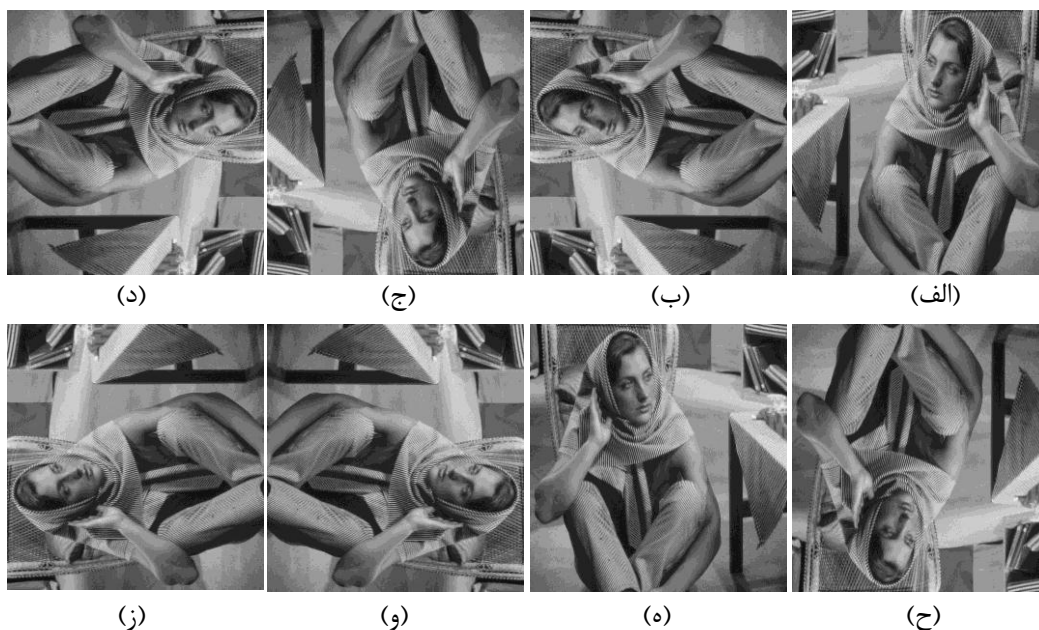
در گام دوم، تصویر حاصل از گام اول، تحت تبدیلات هندسی بازگشت‌پذیر قرار می‌گیرد. این تبدیلات شامل چرخش تصویر در چهار جهت و انعکاس تصویر نسبت به چهار محور می‌باشند. درجات چرخش برابر با ۰، ۹۰، ۱۸۰ و ۲۷۰ در نظر گرفته می‌شود. قابل ذکر است که درجات چرخش دیگر در این روش مناسب نخواهد بود، زیرا منجر به تغییر ابعاد تصویر شده و این امر خود در تغییر حجم تصویر دخیل خواهد بود. نمونه‌ای از این تغییر ابعاد و خرابی در شکل (۴-۴) نشان داده شده است.



(الف) (ب)

شکل ۴-۴: تغییر ابعاد تصویر هنگام چرخش با درجات نامناسب. (الف) تصویر اصلی، (ب) تصویر پس از چرخش ۴۵ درجه.

در شکل (۴-۵) تصاویر حاصل از چرخش تصویر در چهار جهت و تصاویر حاصل از انعکاس تصویر نسبت به چهار محور نشان داده شده‌اند. هر یک از این تصاویر با روش JPEG با نرخ‌های متفاوتی فشرده می‌شوند. علت این است که در روش JPEG، در هر سطر، تفاضل ضریب DC آخرین بلوک از ضریب DC اولین بلوک در سطر بعد، به دست می‌آید (شکل (۴-۶)). بنابراین تبدیلات هندسی، سبب تغییر بلوک‌های ابتدایی و انتهایی هر سطر و در نتیجه تغییر در نرخ فشرده‌سازی خواهند شد.



(الف) (ب) (ج) (د) (ه) (و) (ز) (ح)

شکل ۴-۵: اعمال تبدیلات هندسی بر روی تصویر. (الف) چرخش ۰ درجه، (ب) چرخش ۹۰ درجه، (ج) چرخش ۱۸۰ درجه (د) چرخش ۲۷۰ درجه، (ه) انعکاس تصویر نسبت به محور $x=0$ ، (و) انعکاس تصویر نسبت به محور $y=0$ ، (ز) انعکاس تصویر نسبت به محور $y=x$ ، (ح) انعکاس تصویر نسبت به محور $y=-x$.

DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→
DC	DC	DC	DC	DC	DC	→

شکل ۴-۶: نمایش جهت تفاضل ضرایب DC.

تصاویر حاصل از تبدیلات هندسی، با استفاده از JPEG فشرده شده و حجم هر یک محاسبه می-شود. هر تصویری که حجم کمتری حاصل کند، بهترین تصویر برای اعمال به JPEG خواهد بود. این مرحله از پردازش نیز بازگشت پذیر و بدون اتلاف است، زیرا با انجام معکوس تبدیلات، تصویر به حالت اولیه خود باز می گردد.

۴-۳-۱- مثالی از عملکرد روش پیشنهادی اول

در شکل (۴-۷) تصویر اصلی و تصویری که بلوک های 8×8 آن بر اساس ضرایب DC مرتب شده اند، نشان داده شده است.

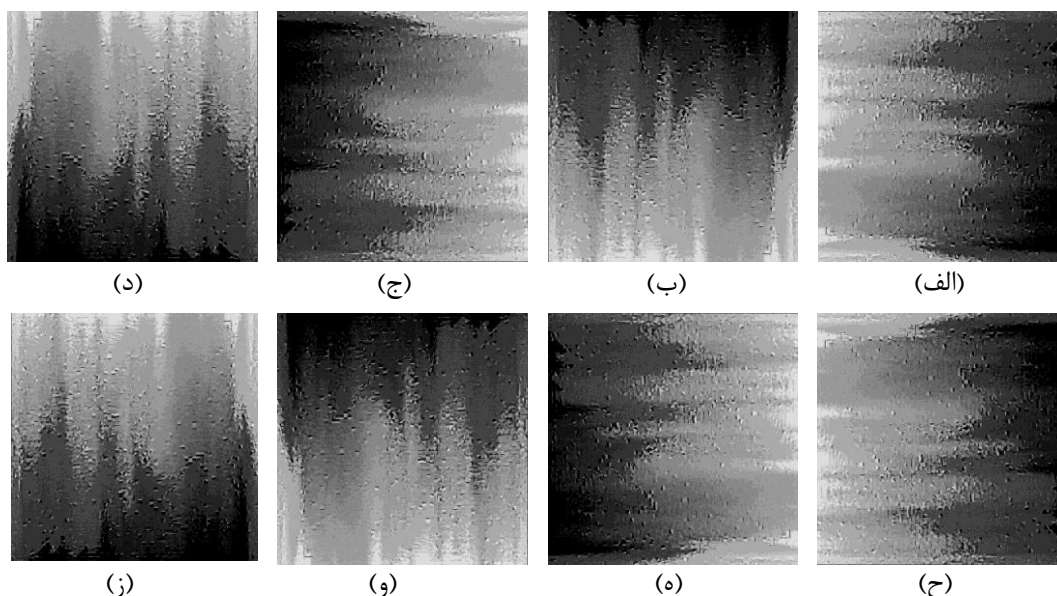


(ب)

(الف)

شکل ۴-۷: نمونه ای از اعمال روش پیشنهادی. (الف) تصویر اصلی، (ب) تصویر پردازش شده.

شکل (۴-۸) تصاویر حاصل از اعمال تبدیلات هندسی بر روی تصویر پردازش شده را نشان می-دهد.



شکل ۴-۸: تصاویر حاصل از اعمال تبدیلات هندسی بر روی تصویر پردازش شده. (الف) چرخش ۰ درجه، (ب) چرخش ۹۰ درجه، (ج) چرخش ۱۸۰ درجه (د) چرخش ۲۷۰ درجه، (ه) انعکاس تصویر نسبت به محور $x=0$ ، (و) انعکاس تصویر نسبت به محور $y=0$ ، (ز) انعکاس تصویر نسبت به محور $y=x$ ، (ح) انعکاس تصویر نسبت به محور $y=-x$.

۴-۴- کاهش تنوع سطوح خاکستری تصویر و متراکم نمودن

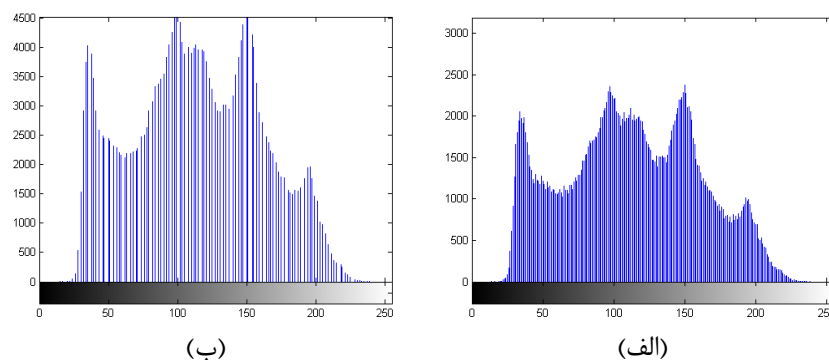
هیستوگرام به منظور کاهش آنتروپی (روش پیشنهادی دوم)

در این بخش یک پیش پردازش شبه بی اتلاف معرفی می شود. در این روش، سعی شده تا تنوع سطوح خاکستری تصویر کاهش پیدا کند. سپس با متراکم نمودن هیستوگرام، سطوح خاکستری در یک محدوده ی نزدیک به هم قرار می گیرند. این پیش پردازش منجر به بهبود عملکرد Lossless JPEG، می شود. تصویر پردازش شده تا حد زیادی قابل بازگشت به تصویر اولیه خواهد بود و اطلاعات کم از دست رفته در برابر میزان بهبود فشرده سازی، ناچیز است.

اولین گام، کم نمودن تنوع سطوح خاکستری تصویر است. با کاهش تنوع، هیستوگرام تصویر پردازش شده، اصطلاحاً خلوت^۱ خواهد شد. برای کاهش تنوع، ابتدا سطوح خاکستری نزدیک به هم از نظر مقدار، بر اساس یک پارامتر ثابت، مشخص شده و در یک دسته قرار می گیرند. سپس، در هر

¹ Sparse

دسته، یک سطح خاکستری به عنوان نماینده‌ی آن دسته تعیین می‌شود. مقدار سطح خاکستری هر پیکسل از تصویر اصلی با مقدار نماینده‌ی مربوط به دسته‌ی خودش، جایگزین می‌گردد. با این عمل، تصویری حاصل می‌شود که تنوع سطوح خاکستری آن، نسبت به تصویر اصلی کمتر است. هر چه مقدار پارامتر بزرگتر باشد، تعداد سطوح قرار داده شده در یک دسته بیشتر و کیفیت تصویر پردازش شده کمتر خواهد بود. هر چه این مقدار کوچکتر باشد، تعداد سطوح موجود در یک دسته کمتر و کیفیت تصویر پردازش شده بیشتر خواهد شد. شکل (۴-۹-الف) هیستوگرام یک تصویر را نشان می‌دهد که با روش مطرح شده به هیستوگرام خلوت شکل (۴-۹-ب) تبدیل شده است. در واقع، برخی از سطوح خاکستری از بین رفته و مقادیر فراوانی برخی از سطوح خاکستری دیگر افزایش یافته است.



شکل ۴-۹: تبدیل هیستوگرام تصویر اصلی به هیستوگرام خلوت. (الف) هیستوگرام تصویر اصلی، (ب) هیستوگرام تصویر پردازش شده.

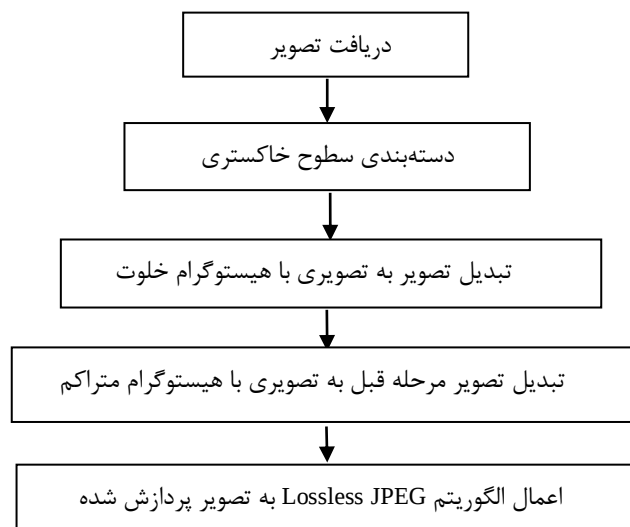
به‌منظور اینکه این روش کمترین تغییر را در تصویر اصلی ایجاد کند، موارد زیر را مدنظر قرار داده-

ایم:

✓ در هر دسته، سطحی به‌عنوان نماینده انتخاب می‌شود که در بین سایر سطوح موجود در آن دسته ماکزیمم مقدار فراوانی را در نمودار هیستوگرام تصویر اصلی داشته باشد. علت انتخاب این سطح به عنوان نماینده، این است که زمانی که مقادیر موجود در یک دسته با این مقدار جایگزین می‌شوند، تعداد بیشتری از سطوح، مقدار قبلی خود را حفظ می‌کنند. یعنی مقدار قبلی آنها برابر با مقدار نماینده‌ی انتخابی است. در نتیجه کمترین تغییرات در مقادیر سطوح، صورت می‌گیرد و دقت افزایش می‌یابد.

✓ در این پیش‌پردازش، مقدار پارامتر تعیین‌کننده‌ی تعداد سطوح هر دسته، کمترین مقدار ممکن و برابر با دو در نظر گرفته شده است.

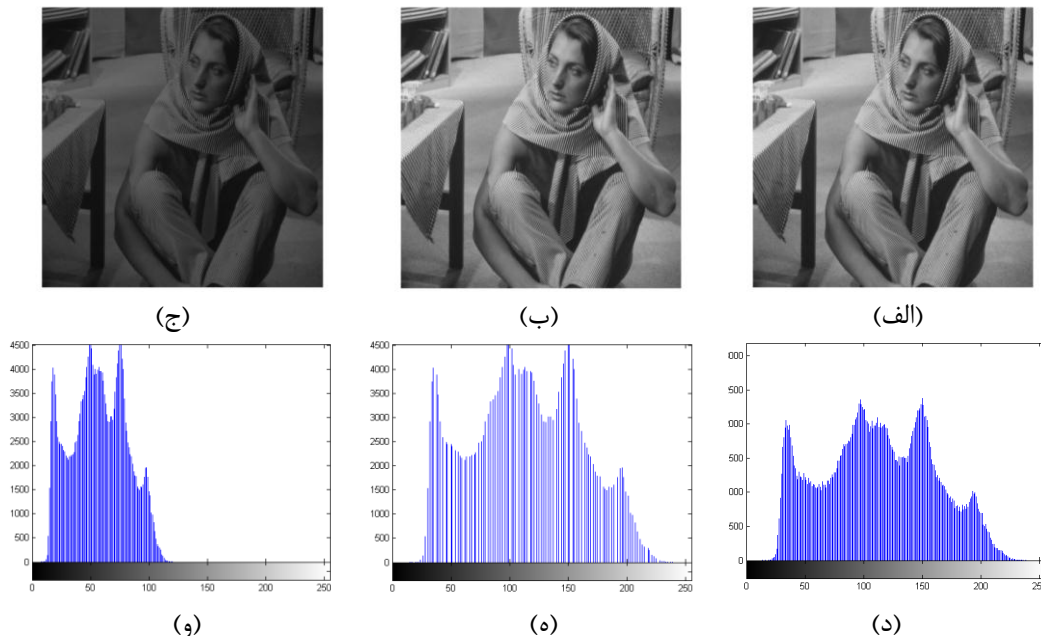
پس از خلوت نمودن هیستوگرام، در گام دوم، هیستوگرام خلوت‌شده، متراکم می‌شود. با این عمل مقادیر پیکسل‌های تصویر به هم نزدیک شده و میزان اختلاف آنها از یکدیگر کمتر خواهد شد. در نتیجه آنتروپی کاهش یافته و افزونگی افزایش می‌یابد و تصویر حاصل با نرخ بیشتری متراکم خواهد شد. شکل (۴-۱۰) بلوک دیاگرام روش پیشنهادی دوم را نشان می‌دهد.



شکل ۴-۱۰: بلوک دیاگرام روش پیشنهادی دوم.

همانطور که در شکل (۴-۱۱) نشان داده شده است، تصویری که هیستوگرام آن متراکم شده، نسبت به تصویر اصلی تفاوت زیادی دارد. لذا باید به تصویر حاصل از گام اول پردازش (که تفاوت محسوسی با تصویر اصلی ندارد) بازیابی شود. اطلاعات مورد نیاز برای بازیابی، شامل مقادیر سطوح خاکستری است که در گام اول، به عنوان نماینده انتخاب شده‌اند. زیرا برای بازیابی تصویر نیاز است که نگاهی یک به یک بین مقادیر سطوح خاکستری تصویر با هیستوگرام متراکم و مقادیر سطوح خاکستری تصویر با هیستوگرام خلوت، انجام گیرد. این روش پیش‌پردازش، نزدیک به بی‌اتلاف بوده و با در نظر گرفتن میزان سربار نیز منجر به بهبود قابل ملاحظه‌ای در عملکرد Lossless JPEG می‌شود.

۴-۴-۱- مثالی از عملکرد روش پیشنهادی دوم



شکل ۴-۱۱: مثالی از عملکرد روش پیشنهادی دوم، (الف) تصویر اصلی، (ب) تصویر با تنوع کمتر سطوح خاکستری، (ج) تصویر با هیستوگرام متراکم، (د) هیستوگرام تصویر (۴-۱۱-الف)، (ه) هیستوگرام تصویر (۴-۱۱-ب)، (و) هیستوگرام تصویر (۴-۱۱-ج).

۴-۵- دسته‌بندی سطوح خاکستری و متراکم کردن هیستوگرام

تصویر بر اساس سیستم بینایی انسان (روش پیشنهادی سوم)

در این روش نیز، ابتدا تصویر به بلوک‌های بدون هم‌پوشانی تقسیم شده و سپس هر بلوک بر اساس سیستم بینایی انسان مورد پردازش قرار می‌گیرد. این روش مشابه روش پیشنهادی سوم است. اما مقدار پارامتر تعیین‌کننده تعداد سطوح در هر دسته، برای هر بلوک از تصویر، متفاوت خواهد بود. این مقدار بر اساس عملکرد سیستم بینایی انسان انتخاب می‌شود. ابتدا به توضیحاتی در مورد عملکرد سیستم بینایی انسان نسبت به تغییرات تصویر، خواهیم پرداخت و در ادامه روش پیشنهادی سوم را شرح خواهیم داد.

۴-۵-۱- عملکرد سیستم بینایی انسان نسبت به تغییرات

سیستم بینایی انسان، نسبت به تغییرات به یک میزان حساسیت نشان نمی‌دهد، عوامل مختلفی بر میزان حساسیت چشم نسبت به تغییرات تصویر تاثیرگذار هستند. در ادامه به توضیح چند عامل مهم خواهیم پرداخت.

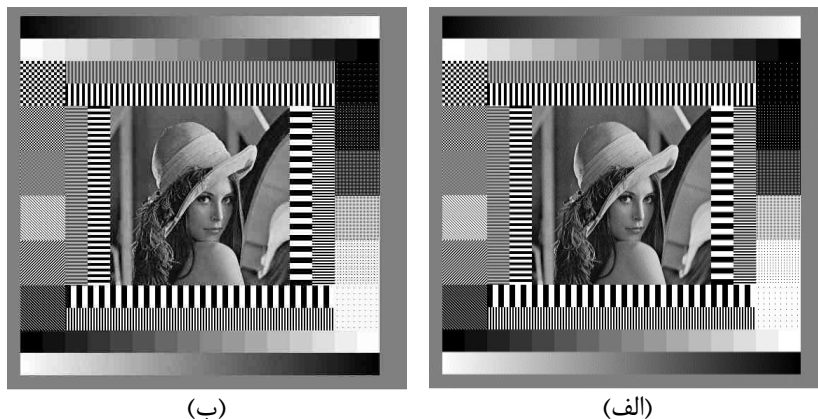
✓ **حساسیت به بافت^۱:** چشم انسان نسبت به بخش‌هایی که بافت ناهمواری دارند، حساسیت کمتری نسبت به تغییرات دارد. به عبارت دیگر تغییرات در بخش‌هایی از تصویر که بافت شلوغی دارند، خیلی کمتر از نواحی با بافت هموار و یکنواخت قابل رویت است (شکل ۴-۱۲).



شکل ۴-۱۲: واکنش متفاوت چشم انسان نسبت به تغییرات در نواحی با بافت متفاوت.

✓ **پوشش:** به افزایش آستانه‌ی تحریک بصری یک سیگنال بر اثر مجاورت با سیگنال دیگر، پدیده‌ی پوشش گفته می‌شود. یک عنصر از تصویر می‌تواند قابلیت آشکارسازی عناصر تصویری مجاور آن را کاهش دهد. وجود لبه در بخشی از تصویر، قابلیت رویت عناصر مجاور آن را کاهش می‌دهد. به عنوان مثال در شکل (۴-۱۳) تصویر اولیه با کیفیت مطلوب و همان تصویر با کیفیت پایین نشان داده شده است. وجود لبه در اطراف تصویر، باعث کاهش قابلیت رویت خرابی در تصویر (۴-۱۳-ب) شده است.

¹ Texture



شکل ۴-۱۳: تاثیر پدیده‌ی پوشش در کاهش قابلیت رویت خرابی تصویر. (الف) تصویر اصلی، (ب) تصویر با خرابی زیاد و کیفیت پایین.

✓ **حساسیت نسبت به شدت روشنایی:** اعوجاج در بخش‌هایی از تصویر که شدت روشنایی خیلی بالا یا شدت روشنایی خیلی پایین دارند، کمتر از نواحی با شدت روشنایی متوسط، قابل تشخیص است (شکل (۴-۱۴)).



شکل ۴-۱۴: حساسیت کم چشم نسبت به خرابی در نواحی بسیار روشن و بسیار تیره.

۴-۵-۲- انتخاب پارامتر تعیین‌کننده‌ی تعداد سطوح هر دسته و دسته‌بندی

سطوح خاکستری

انتخاب پارامتر تعیین‌کننده‌ی تعداد سطوح هر دسته و دسته‌بندی سطوح خاکستری، بر اساس سیستم بینایی انسان صورت می‌گیرد. همانطور که در بخش (۴-۵-۱) گفته شد، عملکرد سیستم بینایی انسان به‌گونه‌ای است که در برخی از بخش‌های تصویر حساسیت کمتری نسبت به خرابی نشان می‌دهد و در برخی نواحی دیگر، نسبت به خرابی از حساسیت بیشتری برخوردار است. در روش

پیشنهادی، بر اساس موقعیت پیکسل‌ها، تصمیم گرفته می‌شود که پارامتر تعیین‌کننده تا چه حد کوچک یا بزرگ باشد. با این عمل محل قرار گرفتن یک پیکسل، در میزان تغییرات آن نقش خواهد داشت. سپسیت برای هر بلوک از تصویر، عمل خلوت کردن هیستوگرام و سپس متراکم نمودن صورت می‌گیرد.

در مرجع [۵۶] تابعی به نام رویت نویز^۱ تعریف شده است. به کمک این تابع، ابتدا میزان مشاهده‌پذیری تغییرات در مقادیر پیکسل بر اساس مقادیر پیکسل‌های همسایه‌اش به دست می‌آید. بنابراین به کمک این تابع، می‌توان تغییرات مجاز را تخمین زد. منظور از میزان تغییرات مجاز، حداکثر میزان تغییرات سطح خاکستری است به گونه‌ای که چشم انسان قادر به تشخیص این تغییرات نباشد. به طور کلی تابع رویت نویز به صورت فرمول (۱-۴) تعریف می‌شود.

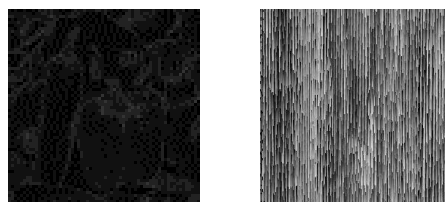
$$NVF(i, j) = \frac{1}{1 + \delta_x^2(i, j)} \quad (1-4)$$

در فرمول (۱-۴)، $\delta_x^2(i, j)$ واریانس محلی تصویر در یک پنجره‌ی $N \times N$ می‌باشد که پیکسل $p(i, j)$ در مرکز آن قرار گرفته است. هر چه میزان NVF به یک نزدیکتر باشد به این معناست که تغییرات در یک پیکسل بیشتر توسط چشم انسان تشخیص داده می‌شود. زیرا پیکسل در قسمت‌های هموار تصویر قرار دارد. هر چه مقدار این تابع کمتر باشد تغییرات نامشهودتر است.

در این روش هر بلوک از تصویر به طور مجزا، مورد پردازش قرار می‌گیرد. لذا برای هر بلوک، باید نگاشتی یک به یک بین مقادیر سطوح خاکستری هیستوگرام متراکم شده با مقادیر سطوح خاکستری هیستوگرام خلوت شده، انجام شود. بر این اساس برای هر بلوک از تصویر، باید اطلاعاتی ذخیره شود. در این روش میزان سربار نسبت به روش پیشنهادی دوم بیشتر است، اما میزان بهبود عملکرد الگوریتم فشرده‌سازی افزایش می‌یابد. سربار شامل سه بخش است: بخش اول، دربردارنده‌ی مقادیر نماینده‌ی هر دسته در هر بلوک، بخش دوم، دربردارنده‌ی تعداد نماینده‌های (دسته‌های) موجود در هر بلوک و بخش سوم، شامل مقادیر تعیین‌کننده‌ی تعداد سطوح خاکستری برای هر دسته در هر بلوک،

¹ Noise Visibility Function

می‌باشد. با تغییر اندازه‌ی بلوک‌های تصویر، میزان سربار ایجاد شده در هر بخش نیز، تغییر خواهد کرد. لذا باید تعادلی بین این دو برقرار نمود. به همین علت، بلوک‌بندی تصویر، با ابعاد مختلف انجام شده و بهترین سایز بلوک‌بندی بر اساس میزان سربار و میزان نرخ فشرده‌سازی، انتخاب می‌شود. در این روش نیز مقادیر سربار را در قالب تصاویر با فرمت PNG ذخیره می‌کنیم. در شکل (۴-۱۵) تصاویر دربردارنده‌ی مقادیر سربار نشان داده شده است.



(ب)

(الف)

شکل ۴-۱۵: تصاویر در بردارنده‌ی اطلاعات لازم برای بازیابی. (الف) تصویر دربردارنده‌ی مقادیر و تعداد نماینده‌ی هر دسته در هر بلوک (بلوک‌بندی 8×8 تصویر اولیه). (ب) تصویر دربردارنده‌ی مقادیر تعیین‌کننده تعداد نماینده‌ها در هر بلوک (بلوک‌بندی 8×8 تصویر اولیه).

این پیش‌پردازش با حفظ کیفیت دیداری از نظر چشم انسان صورت گرفته است و منجر به بهبود $21/37$ درصدی عملکرد Loosless JPEG و بهبود $23/33$ درصدی عملکرد JPEG2000 شده است.

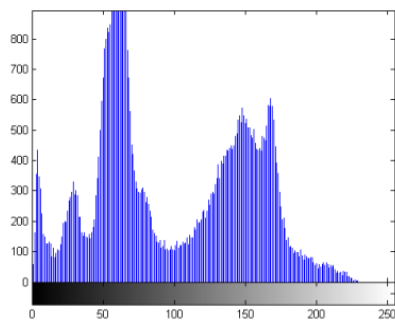
۴-۵-۳- مثالی از عملکرد روش پیشنهادی سوم

تصاویر (۴-۱۶) تا (۴-۱۸)، مراحل روش پیشنهادی را بر روی بلوک‌های یک تصویر، نشان می‌-

دهند.



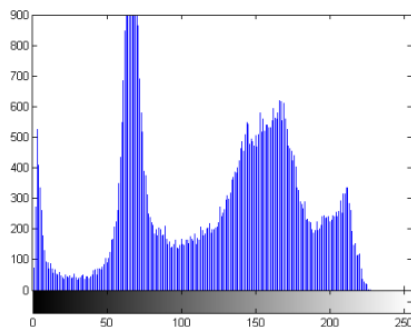
(ب)



(د)



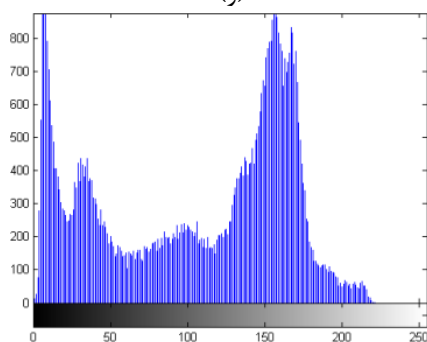
(الف)



(ج)



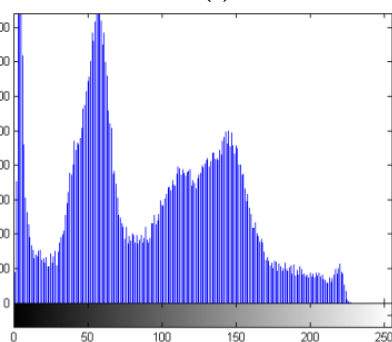
(و)



(ح)



(ه)



(ز)

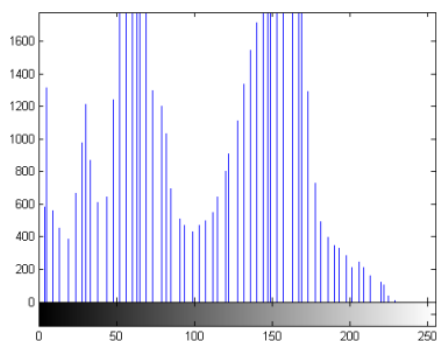
شکل ۴-۱۶: بلوک‌های 256×256 تصویر اصلی به همراه هیستوگرام هر بلوک. (الف) بلوک اول، (ب) بلوک دوم، (ج) هیستوگرام بلوک اول، (د) هیستوگرام بلوک دوم. (ه) بلوک سوم، (و) بلوک چهارم، (ز) هیستوگرام بلوک سوم، (ح) هیستوگرام بلوک چهارم.



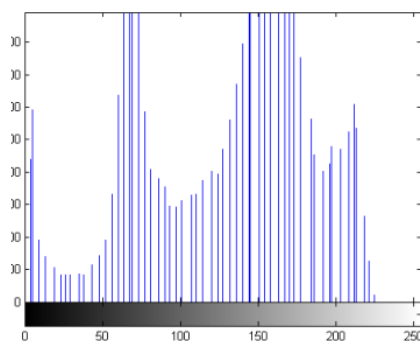
(ب)



(الف)



(د)



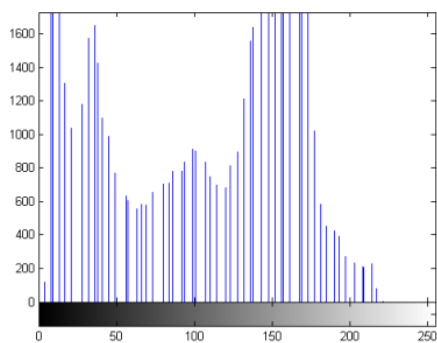
(ج)



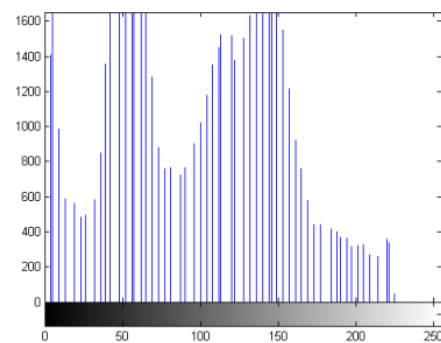
(و)



(ه)

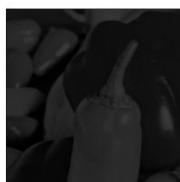


(ح)

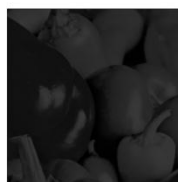


(ز)

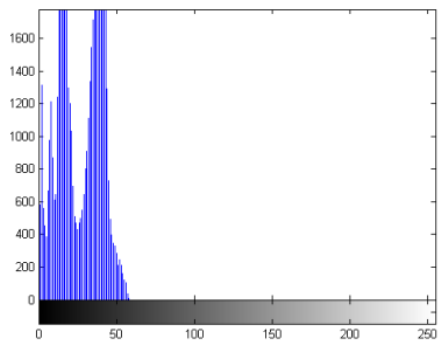
شکل ۱۷-۴: شکل ۱۷-۴: بلوک‌های تصویر پردازش شده به همراه هیستوگرام هر بلوک. (الف) بلوک اول، (ب) بلوک دوم، (ج) هیستوگرام بلوک اول، (د) هیستوگرام بلوک دوم (ه) بلوک سوم، (و) بلوک چهارم، (ز) هیستوگرام بلوک سوم، (ح) هیستوگرام بلوک چهارم.



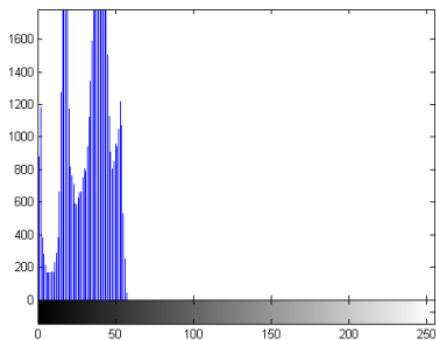
(ب)



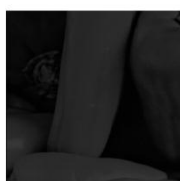
(الف)



(د)



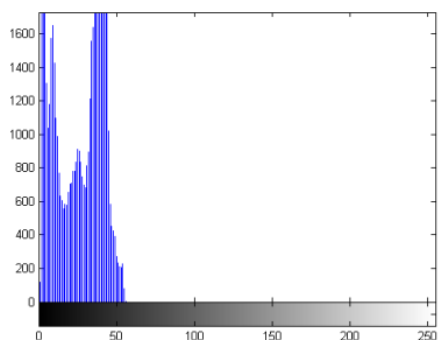
(ج)



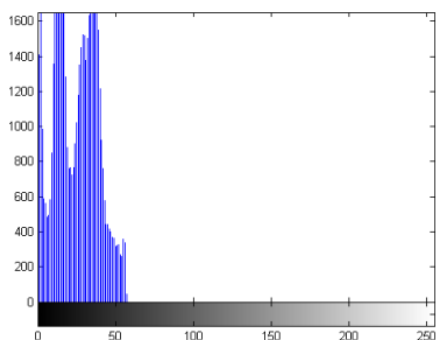
(و)



(ه)



(ح)



(ز)

شکل ۴-۱۸: بلوک‌های تصویر با مقادیر سطوح خاکستری نزدیک به هم به همراه هیستوگرام هر بلوک. (الف) بلوک اول، (ب) بلوک دوم، (ج) هیستوگرام بلوک اول، (د) هیستوگرام بلوک دوم. (ه) بلوک سوم، (و) بلوک چهارم، (ز) هیستوگرام بلوک سوم، (ح) هیستوگرام بلوک چهارم.

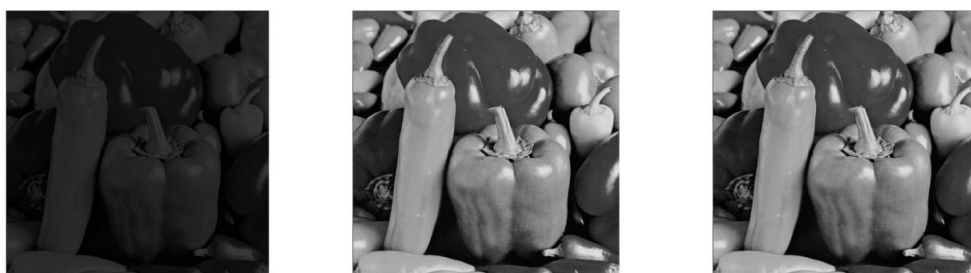
در شکل (۴-۱۹) تصاویر در بردارنده‌ی اطلاعات لازم برای بازیابی، نشان داده شده است. (قابل ذکر است که ابعاد واقعی این تصاویر از آنچه در شکل نشان داده شده، کوچکتر هستند).



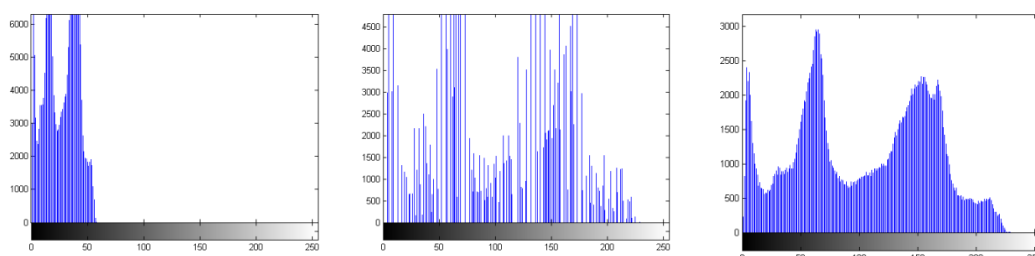
(الف) (ب)

شکل ۴-۱۹: تصاویر در بردارنده‌ی اطلاعات لازم برای بازیابی. (الف) تصویر دربردارنده‌ی مقادیر و تعداد نماینده هر دسته در هر بلوک از تصویر پردازش شده، (ب) تصویر دربردارنده‌ی مقادیر تعیین‌کننده تعداد نماینده‌ها در هر بلوک از تصویر پردازش شده.

در شکل (۴-۲۰) نتیجه‌ی عملکرد روش مطرح شده بر روی یک تصویر نشان داده شده است.



(الف) (ب) (ج)



(الف) (ب) (ج)

شکل ۴-۲۰: نمونه‌ای از اعمال روش پیشنهادی چهارم بر روی تصویر pepper. (الف) تصویر اولیه، (ب) تصویر با بلوک‌هایی با هیستوگرام خلوت، (ج) تصویر با بلوک‌هایی با هیستوگرام متراکم، (د) هیستوگرام شکل (۴-۲۰-الف)، (و) هیستوگرام شکل (۴-۲۰-ب)، (ه) هیستوگرام شکل (۴-۲۰-ج).

۴-۶- یافتن فضای رنگ مناسب برای هر تصویر به منظور افزایش

نرخ فشرده‌سازی (روش پیشنهادی چهارم)

همانطور که در بخش (۲-۵) گفته شد، یک فضای رنگ مناسب برای فشرده‌سازی، باید دارای چند ویژگی باشد. در حال حاضر، فضاهای رنگ متفاوتی برای نمایش تصویر، وجود دارند، اما هیچکدام منحصر برای فشرده‌سازی تصویر ایجاد نشده‌اند. یافتن یک فضای رنگ بهینه که منجر به نرخ فشرده-

سازی بیشتری شود، نیاز به جستجو در یک فضای گسترده دارد. لذا با استفاده از الگوریتم ژنتیک می‌توان این فضای رنگ بهینه را به‌دست آورد.

در این روش سعی داریم برای هر تصویر، با استفاده از الگوریتم ژنتیک، ماتریس نگاشت مناسب را یافته و بر اساس این ماتریس نگاشت، تصویر را به فضای رنگ مناسب برای فشرده‌سازی انتقال دهیم. با این عمل، الگوریتم‌های فشرده‌سازی عملکرد بهتری در فشرده‌سازی تصویر حاصل، خواهند داشت [۵۷-۵۸]. توضیحات بیشتر در رابطه با الگوریتم ژنتیک در پیوست (۱) آمده است. در ادامه به توضیح ساختار الگوریتم ژنتیک در روش پیشنهادی، می‌پردازیم.

۴-۶-۱- ساختار کروموزوم

همانطور که در فرمول (۲-۴) نشان داده شده است، در این پیش‌پردازش به دنبال اعدادی هستیم که با ضرب در مقادیر سه کانال قبلی، منجر به تولید سه کانال جدید شده و این سه کانال در کنار هم یک فضای رنگ مناسب برای فشرده‌سازی را ایجاد کنند. برای ایجاد هر کانال جدید، نیاز به سه ضریب داریم و در کل برای ایجاد یک فضای رنگ جدید نیازمند یافتن نه ضریب هستیم. بنابراین کروموزوم در این تکنیک، مطابق شکل (۴-۲)، شامل نه ژن است.

a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}
----------	----------	----------	----------	----------	----------	----------	----------	----------

شکل ۴-۲: ساختار کروموزوم در تکنیک پیشنهادی.

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (۲-۴)$$

در فرمول (۲-۴)، با ضرب ماتریس نگاشت در هر یک از کانال‌های R ، G و B سه کانال جدید X_1 ،

X_2 و X_3 ایجاد می‌شوند. هر یک از عناصر ماتریس نگاشت، یک ژن از کروموزوم را تشکیل می‌دهند.

۴-۶-۲- تابع برازش

لازمه‌ی حل هر مسئله با استفاده از الگوریتم‌های ژنتیکی، تعریف یک تابع برازش مناسب می‌باشد. این تابع، نشان‌دهنده‌ی شایستگی یا برازندگی هر کروموزوم است. در واقع باید به نحوی کروموزوم‌ها را مورد سنجش قرار داده و آنها را برای ورود به نسل بعدی انتخاب کرد و یا اینکه آنها را حذف نمود. برای این سنجش، از تابع برازش استفاده می‌کنیم.

در مسئله‌ی مورد نظر، معیار سنجش، کاهش حجم است. به دلیل اینکه چشم انسان نسبت به اطلاعات روشنایی، حساس‌تر از اطلاعات رنگ می‌باشد، باید مقادیر به گونه‌ای انتخاب شوند که اکثر اطلاعات روشنایی در یک کانال و بخش کمی از اطلاعات در دو کانال دیگر قرار بگیرند. از طرف دیگر با توجه به مطالب گفته شده در بخش (۲-۵)، فضای رنگی مناسب است که در آن وابستگی بین کانال‌های تصویر کم باشد. برای به‌دست آوردن سه کانال جدید از فرمول‌های (۴-۳) تا (۴-۵) استفاده می‌شود [۵۹].

$$T_1 = a_{11} \times R + a_{12} \times G + a_{13} \times B \quad (۴-۳)$$

$$T_2 = a_{21} \times R + a_{22} \times G + a_{23} \times B \quad (۴-۴)$$

$$T_3 = a_{31} \times R + a_{32} \times G + a_{33} \times B \quad (۴-۵)$$

انرژی هر یک از کانال‌های تصویر، بر اساس فرمول‌های (۴-۶) تا (۴-۸) به‌دست می‌آید [۶۰].

$$T_1 E = 100 \times \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_1^2 ij}{TE_{T_1 T_2 T_3}} \quad (۴-۶)$$

$$T_2 E = 100 \times \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_2^2 ij}{TE_{T_1 T_2 T_3}} \quad (۴-۷)$$

$$T_3 E = 100 \times \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_3^2 ij}{TE_{T_1 T_2 T_3}} \quad (۴-۸)$$

انرژی کل تصویر نیز به صورت فرمول (۴-۹) محاسبه می‌گردد.

$$TE_{T_1 T_2 T_3} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_1^2 + \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_2^2 + \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_3^2 \quad (۴-۹)$$

با توجه به مطالب گفته شده، سعی می‌شود که اکثر انرژی در یک کانال از تصویر تجمع داشته باشد. لذا با ماکزیمم شدن فرمول (۴-۱۰) این امر محقق می‌شود.

$$f_1 = T_1E - (T_2E + T_3E) \quad (۴-۱۰)$$

همچنین سعی می‌شود که وابستگی بین کانال‌های تصویر کم باشد. با مینیمم شدن مقادیر فرمول‌های (۴-۱۱) تا (۴-۱۲) می‌توان وابستگی بین هر دو کانال را کاهش داد.

$$f_2 = correlation(T_1, T_2) \quad (۴-۱۱)$$

$$f_3 = correlation(T_1, T_3) \quad (۴-۱۲)$$

$$f_4 = correlation(T_2, T_3) \quad (۴-۱۳)$$

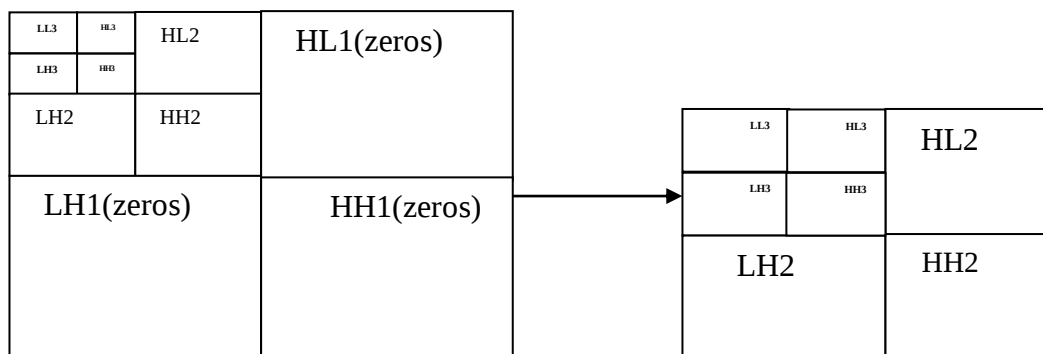
بنابراین، مقادیر f_1, f_2, f_3 و f_4 به‌عنوان توابع برازش معرفی می‌شوند.

۴-۶-۳- الگوریتم فشرده‌سازی مورد استفاده برای روش پیشنهادی چهارم

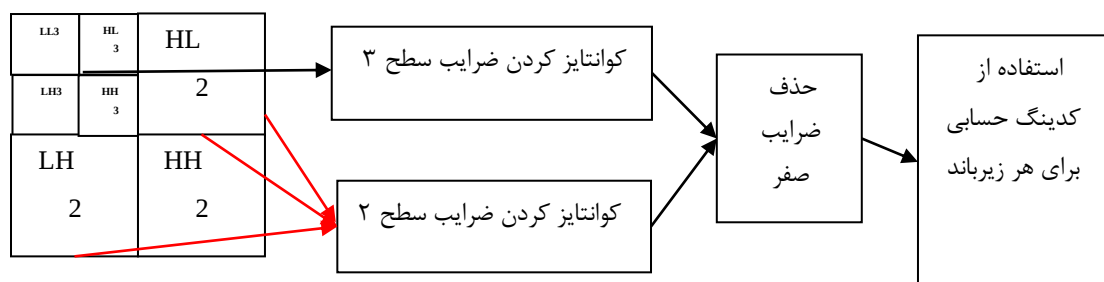
پس از انجام مراحل پیش‌پردازش، نیاز است که تصویر با استفاده از الگوریتمی فشرده شود. الگوریتم‌هایی وجود دارند که در آغاز مرحله‌ی فشرده‌سازی، تصویر را از فضای رنگ RGB به فضای رنگ YCBCR تبدیل می‌کنند. هدف این است که در همین مرحله از الگوریتم فشرده‌سازی، تصویر به‌جای تبدیل به فضای رنگ YCBCR به فضای رنگ به دست آمده از روش پیشنهادی چهارم، تبدیل شود. در این پژوهش الگوریتم فشرده‌سازی مورد استفاده،^۱ CIC است. این تکنیک یک الگوریتم ترکیبی است که در سال ۲۰۱۱ توسط Siddeq برای فشرده‌سازی تصاویر رنگی، طراحی و ارائه شد. در این روش از ترکیب تبدیل ویولت و تبدیل گسسته‌ی یک بعدی استفاده شده است. در ابتدا بر روی تصویر تا سه سطح تبدیل ویولت صورت می‌گیرد. در مرحله‌ی بعد ضرایب زیرباند دوم و سوم، کوانتایز شده و پس از حذف مقادیر صفر، با استفاده از کدینگ حسابی کد می‌شوند. در مرحله‌ی آخر نیز تبدیل گسسته کسینوسی بر روی مقادیر LL3 اعمال شده و مقادیر کوانتایز می‌شوند. اطلاعات در

^۱ Color Image Compression

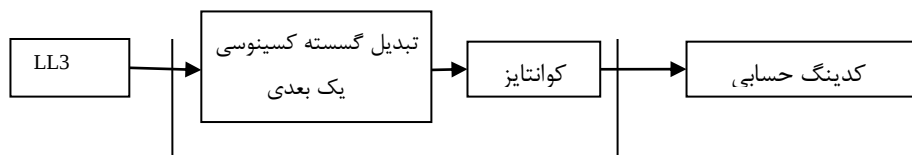
قالب یک فایل با پسوند CIC ذخیره می‌گردند. مراحل این الگوریتم در شکل (۴-۲۲) نشان داده شده است.



(الف)



(ب)



(ج)

شکل ۴-۲۲: مراحل فشرده‌سازی CIC. (الف) تبدیل ویولت، (ب) کوانتایز کردن و حذف ضرایب صفر در هر زیرباند و استفاده از کدینگ حسابی، (ج) فشرده‌سازی LL3 با استفاده از تبدیل گسسته‌ی کسینوسی.

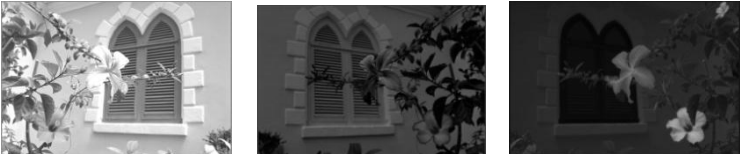
۴-۶-۴- مثال از عملکرد روش پیشنهادی چهارم

در شکل (۴-۲۳) چند تصویر در فضای RGB و همان تصاویر در فضای رنگ به‌دست آمده با روش پیشنهادی، نشان داده شده‌اند. جدول (۴-۱) نیز کانال‌های هر تصویر را نشان می‌دهد.



شکل ۴-۲۳: تغییر فضای رنگ تصاویر با استفاده از روش پیشنهادی چهارم. تصاویر سمت چپ: تصاویر در فضای رنگ RGB، تصاویر سمت راست: تصاویر در فضای رنگ جدید

جدول ۴-۱: عملکرد روش پیشنهادی چهارم بر روی کانال‌های چند نمونه از تصاویر.

کانال‌های تصویر قبل از پردازش	کانال‌های تصویر پس از پردازش
	
	
	
	

همان‌طور که در جدول (۴-۱) نشان داده شده است، پس از پردازش، اکثر اطلاعات تصویر، در یک کانال تجمع می‌یابد. همین امر امکان فشرده‌سازی با نرخ بیشتر را فراهم می‌کند.

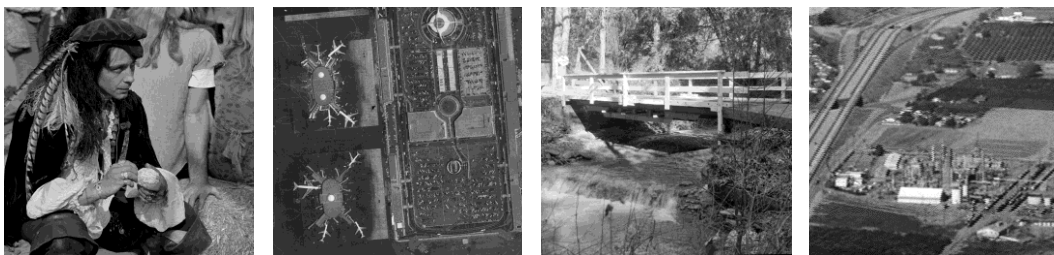
۴-۷-ارزیابی روش‌های پیشنهادی

برای انجام آزمایشات و به‌دست آوردن نتایج، نیازمند پایگاه داده‌ای استاندارد از تصاویر هستیم. همچنین برای ارزیابی روش‌های ارائه شده، نیازمند معیارهای سنجش کیفیت تصویر و سنجش میزان فشرده‌سازی خواهیم بود. در این بخش ابتدا به بررسی و بیان این مفاهیم، خواهیم پرداخت. سپس نتایج حاصل از اعمال روش‌های پیشنهادی را در بهبود هر یک از روش‌های فشرده‌سازی مورد نظر، نشان می‌دهیم. در انتها نتایج حاصل از روش پیشنهادی را با مراجع دیگر مورد مقایسه قرار خواهیم داد. قابل ذکر است که کلیه‌ی آزمایشات، با استفاده از نرم‌افزار Matlab نسخه R20012a و بر روی سیستم اینتل با پردازنده‌ی ۲/۶ GHZ و حافظه‌ی ۴ گیگا بایت بررسی و پیاده‌سازی شده است.

۴-۷-۱-توصیف پایگاه داده^۱ USD

در این پایان‌نامه برای بررسی روش‌های پیشنهادی اول تا سوم، از پایگاه داده USD استفاده نموده- ایم [۶۰]. این پایگاه داده، در مجموع دارای ۴۴ تصویر می‌باشد. ۱۴ تصویر با ابعاد ۲۵۶×۲۵۶، ۲۶ تصویر با ابعاد ۵۱۲×۵۱۲ و چهار تصویر با ابعاد ۱۰۲۴×۱۰۲۴ در این پایگاه داده قرار دارند. این مجموعه شامل تصاویر مختلفی اعم از تصویر چهره انسان، منظره، تصاویری با پس‌زمینه ساده و تصاویری با جزئیات کم و جزئیات زیاد است. در شکل (۴-۲۴) چهار نمونه از تصاویر این پایگاه داده نشان داده شده است.

¹ University Of Southern California



شکل ۵۴-۲۴: چهار نمونه تصویر از تصاویر پایگاه داده USD.

۴-۷-۲-توصیف پایگاه داده Kodak

برای بررسی روش پیشنهادی چهارم از پایگاه داده Kodak استفاده نموده‌ایم [۶۱]. این پایگاه داده در مجموع دارای ۲۴ تصویر رنگی در ابعاد ۷۶۸×۵۱۲ ، یا ۵۱۲×۷۶۸ می‌باشد. شکل (۴-۲۵) تصاویر این پایگاه داده را نشان می‌دهد.



شکل ۴-۲۵: تصاویر پایگاه داده Kodak.

۴-۷-۳- معیارهای سنجش الگوریتم‌های فشرده‌سازی

میزان کارآمدی یک روش فشرده‌سازی، با استفاده از معیارهای مختلفی سنجیده می‌شود. یک دسته از این معیارها، میزان فشرده‌سازی تصویر را مورد توجه قرار می‌دهند و دسته‌ی دیگر، کیفیت تصویر بازیابی شده، را ارزیابی می‌کنند. در واقع اگر روش فشرده‌سازی، منجر به از دست رفتن بخشی از اطلاعات شود، باید با استفاده از معیاری، میزان خرابی تصویر را نسبت به تصویر اصلی نشان داد. در ادامه چند معیار مهم از هر دسته را معرفی خواهیم کرد.

۴-۷-۳-۱- معیارهای سنجش میزان فشرده‌سازی

یکی از شناخته شده‌ترین پارامترها، نرخ بیتی، می‌باشد که بیانگر میانگین تعداد بیت‌ها بر تعداد پیکسل‌های^۱ تصویر است.

$$BPP = \frac{S'}{S} k \quad (14-4)$$

در فرمول (۱۴-۴)، S حجم تصویر اصلی، S' حجم تصویر فشرده شده و k نرخ بیت تصویر اولیه است. زمانی که نرخ بیتی بسیار کم است، معیار نرخ فشرده‌سازی می‌تواند کارآمدتر باشد. این معیار به صورت زیر تعریف می‌شود.

$$CR = \frac{S}{S'} \quad (15-4)$$

در فرمول (۱۵-۴)، S حجم تصویر اصلی، S' حجم تصویر فشرده شده است. گاهی اوقات نیز از معیار فضای ذخیره شده مطابق فرمول (۱۶-۴) به جای دو معیار قبلی، استفاده می‌شود.

$$CR = \left(1 - \frac{S}{S'}\right) \times 100 \quad (16-4)$$

در فرمول (۱۶-۴)، S حجم تصویر اصلی، S' حجم تصویر فشرده شده است. زمانی که یک فایل با حجم اولیه ۱۰ مگابایت فشرده شده و به حجم دو مگابایت برسد، فضای ذخیره شده برابر با ۰/۸ خواهد بود. این مقدار معمولاً بر حسب درصد (۰/۸۰) بیان می‌شود.

¹ Bit Per Pixel

۴-۷-۳-۲- معیارهای ارزیابی کیفیت تصویر

زمانی که فشرده‌سازی با اتلاف باشد، برای ارزیابی کیفیت تصویر بازیابی شده، از معیارهای PSNR

^۱ و MSE ^۲ (فرمول (۱۷-۴) و فرمول (۱۸-۴) استفاده می‌شود [۶۲].

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (17-4)$$

در فرمول (۱۷-۴)، m و n تعداد سطرها و ستون‌های تصویر است. $I(i,j)$ پیکسل تصویر اصلی

است که در سطر i ام و ستون j ام قرار دارد. $K(i,j)$ نیز پیکسل تصویر بازیابی شده است که در سطر

i ام و ستون j ام قرار گرفته است.

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2} \right) \quad (18-4)$$

در (۱۸-۴)، m و n تعداد سطر و ستون تصویر است. $I(i,j)$ ، پیکسل (i,j) از تصویر اصلی است.

$K(i,j)$ نیز پیکسل (i,j) از تصویر بازیابی شده می‌باشد. همچنین در این رابطه، MAX، بیشترین

مقدار ممکن برای یک پیکسل در تصویر است. زمانی که یک پیکسل با هشت بیت نمایش داده می‌-

شود، این مقدار برابر با ۲۵۵ خواهد بود. هرچه تصویر بازیابی‌شده، کیفیت بالاتری داشته باشد،

میزان MSE کمتر و PSNR بالاتری خواهد داشت [۶۳].

معیارهای MSE و PSNR از پرکاربردترین و ساده‌ترین معیارهای سنجش شباهت دو تصویر

هستند. اما این معیارها با ویژگی سیستم بینایی انسان تطابق خوبی ندارند [۶۴-۶۵]. دو تصویر با

مقدار MSE یکسان ممکن است خطاهای متفاوتی داشته باشند. طوری که خطاهای یکی قابل مشاهده

توسط چشم انسان بوده و دیگری توسط چشم قابل مشاهده نباشد. در سال‌های اخیر تلاش‌های زیادی

برای ایجاد روش‌های ارزیابی کیفیت تصویر، بر اساس اصلاح معیار MSE انجام شده است. بر این اساس

شاخص دیگری مبتنی بر شباهت ساختاری بین دو تصویر، معرفی شده است. این معیار برای سنجش

^۱ Peak Signal _to_Noise_ Ratio

^۲ Mean Squared Error

شباهت بین دو تصویر X و Y ، سه عامل روشنایی، کنتراست و ساختار دو تصویر را مقایسه می‌کند. این مقایسه به صورت محلی و در بلوک‌های متناظر برای هر تصویر انجام می‌شود. اگر x و y بلوک‌های محلی متناظر در دو تصویر X و Y باشند، تابع مقایسه شدت روشنایی $l(x,y)$ ، تابع مقایسه کنتراست $C(x,y)$ و تابع مقایسه ساختار $S(x,y)$ به صورت فرمول‌های (۴-۱۹) تا (۴-۲۱) تعریف می‌شوند.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (۴-۱۹)$$

$$C(x,y) = \frac{2\delta_x\delta_y + C_2}{\delta_x^2 + \delta_y^2 + C_2} \quad (۴-۲۰)$$

$$S(x,y) = \frac{\delta_{xy} + C_3}{\delta_x\delta_y + C_3} \quad (۴-۲۱)$$

C_1 ، C_2 و C_3 مقادیر ثابتی هستند که از ناپایداری معادلات هنگامی که مخرج کسر مقدار کوچکی باشد، جلوگیری می‌کنند. μ_x و μ_y مقادیر متوسط در بلوک‌های x و y هستند. δ_x^2 و δ_y^2 نمایانگر مقادیر واریانس و کواریانس بین x و y می‌باشند. شاخص شباهت ساختاری بین دو بلوک x و y از حاصلضرب سه تابع مذکور به صورت فرمول (۴-۲۲) محاسبه می‌شود.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\delta_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\delta_x^2 + \delta_y^2 + C_2)} \quad (۴-۲۲)$$

در نهایت شباهت ساختاری کل، بین دو تصویر X و Y با میانگین‌گیری از مقادیر $SSIM$ ^۱ به دست می‌آید. M نشان‌دهنده‌ی تعداد بلوک‌های تصویر است. در فرمول (۴-۲۳) x_i و y_i بلوک محلی i ام در دو تصویر X و Y را نشان می‌دهند.

$$MSSIM(X,Y) = \frac{1}{M} \sum_{i=1}^M (SSIM(x_i, y_i)) \quad (۴-۲۳)$$

بر این اساس در روش پیشنهادی سوم که بر مبنای سیستم بینایی انسان پیشنهاد شده است، برای مقایسه‌ی کیفیت تصویر پردازش شده با تصویر اصلی، از فرمول (۴-۲۳) استفاده شده که مقدار آن در بازه‌ی (۱، ۱-) می‌باشد. مقدار یک بیانگر ماکزیمم شباهت بین دو تصویر است. فرمول (۴-۲۳) نشان

¹ Structural Similarity Index Measure

می‌دهد که هرگونه تغییر در تصویر، ناشی از سه پارامتر مختلف کاهش همبستگی بین دو تصویر، تغییرات روشنایی دو تصویر و تغییرات اختلاف رنگ سطوح با یکدیگر خواهد بود. فرمول (۴-۱۹) بیانگر ضریب همبستگی بین دو تصویر می‌باشد که مقداری در بازه‌ی (۱،۱-) خواهد داشت. فرمول (۴-۲۰) نشان می‌دهد که تا چه حد میزان متوسط روشنایی دو تصویر به هم نزدیک بوده و مقدار آن در بازه‌ی (۱،۰) خواهد بود. فرمول (۴-۲۱) در بازه‌ی (۱،۰) تعریف می‌شود و تخمینی برای تمایز رنگ در دو تصویر است [۶۶].

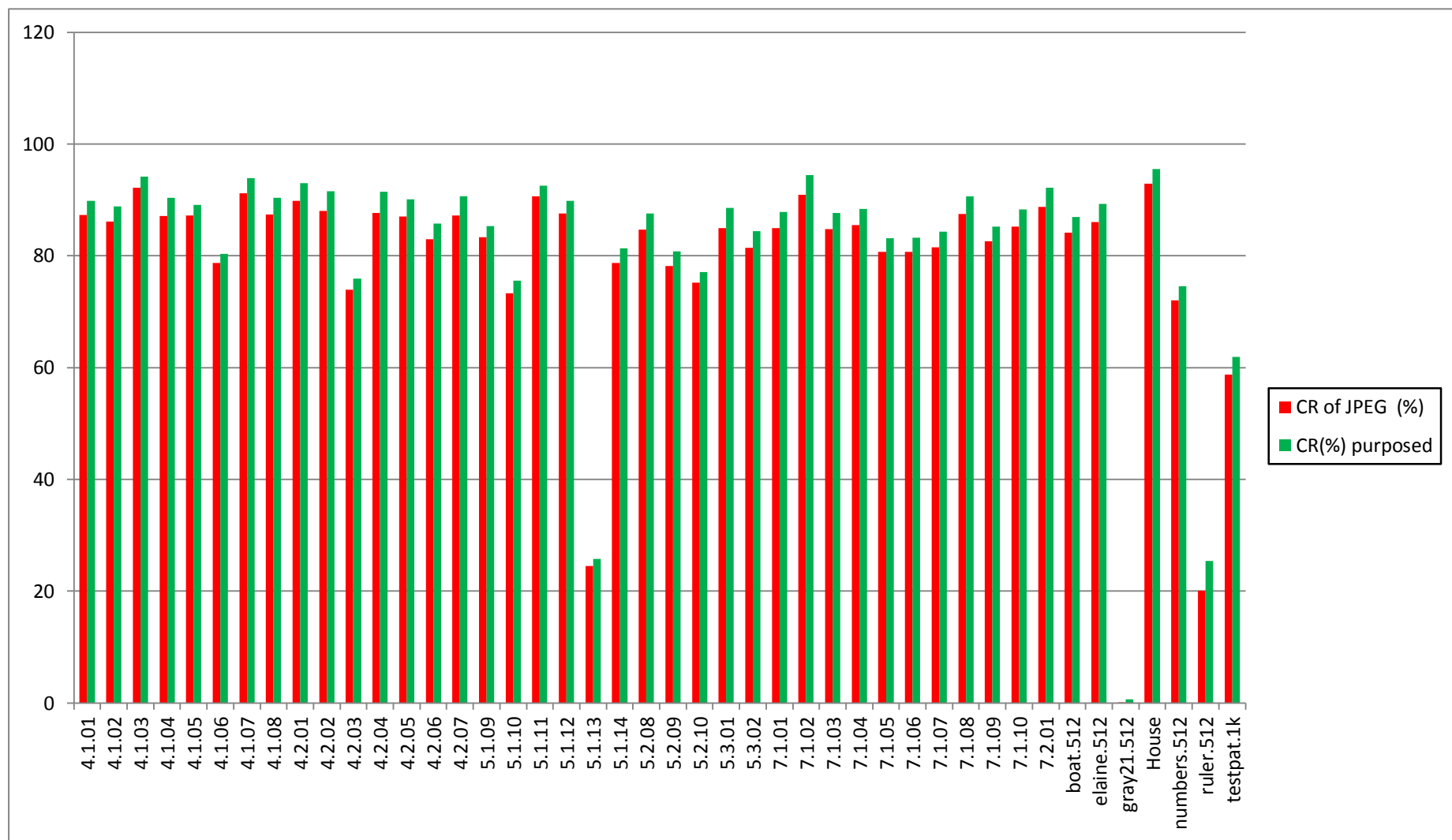
۴-۷-۴- نتایج حاصل از روش پیشنهادی اول بر روی تصاویر USD

در جدول (۴-۲) نتایج حاصل از روش پیشنهادی اول، بر روی تصاویر پایگاه داده USD آورده شده است. همانطور که مشاهده می‌شود، اعمال این پیش‌پردازش باعث افزایش نرخ فشرده‌سازی می‌شود. به هر یک از تبدیلات هندسی، عددی بین یک تا هشت اختصاص داده شده و شماره‌ی تبدیلی که بهترین نرخ فشرده‌سازی را حاصل کند، در جدول مشخص شده است. در تمام موارد پارامتر کیفیت تصویر در الگوریتم فشرده‌سازی JPEG برابر با ۷۵ درصد می‌باشد.

این روش به‌طور میانگین نرخ فشرده‌سازی JPEG را برای تصاویر USD به میزان ۲/۷۲ درصد بهبود داده است. در بهترین حالت میزان بهبود برابر با ۵/۳۳ درصد و در بدترین حالت میزان بهبود برابر با ۱/۶۱ درصد می‌باشد. شکل (۴-۲۶) نمودار میله‌ای مربوط به جدول (۴-۲) را نشان می‌دهد.

جدول ۴-۲: نتایج روش پیشنهادی اول بر روی تصاویر پایگاه داده USD.

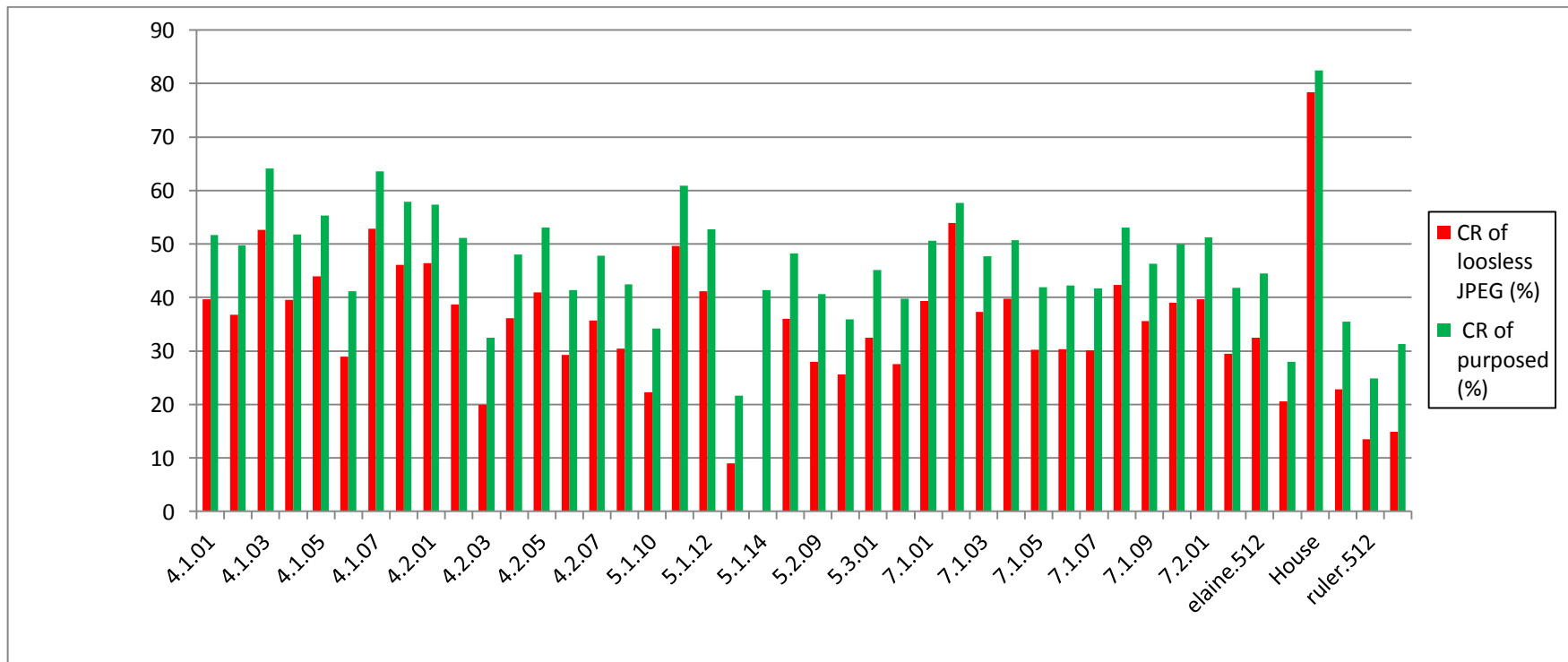
All database	Original size (Kb)	CR(%)	Size (jpg)(Kb)	Purposed (Kb)	No. transform	Improve (%)
4.1.01	۶۴/۴۸	۸۷/۲۹۵۴	۸/۲۱	۸	۲	۲/۵۳
4.1.02	۶۰/۵۸	۸۶/۱۳۵۲	۸/۴	۸/۱۷	۴	۲/۷۰
4.1.03	۶۳/۴۹	۹۲/۲۲۸۸	۴/۹۳۴۶	۴/۸۳	۱	۱/۹۷
4.1.04	۶۴/۷۸	۸۷/۱۶۳۵	۸/۳۱۶۴	۸/۰۴	۲	۳/۲۶
4.1.05	۶۴/۵۳	۸۷/۱۸۱۵	۸/۲۷۲۵	۸/۱۱	۱	۱/۹۷
4.1.06	۶۳/۵۰	۷۸/۷۱۵۳	۱۳/۵۱	۱۳/۲۸	۳	۱/۶۱
4.1.07	۵۳/۹۹	۹۱/۲۳۷۴	۴/۷۳	۴/۶۰	۲	۲/۶۶
4.1.08	۵۷/۷۲	۸۷/۳۹۴۷	۶/۴۶۶۸	۶/۲۷	۴	۲/۹۵
4.2.01	۲۵۵/۹۹	۸۹/۸۸۳۳	۲۵/۸۹۶۴	۲۵/۰۸	۴	۳/۱۳
4.2.02	۲۵۶/۳۳	۸۸/۰۲۵۶	۳۰/۶۹	۲۹/۵۹	۱	۳/۵۷
4.2.03	۲۵۸/۵۶	۷۳/۹۶۹۹	۶۷/۳۰۵۷	۶۵/۹۹	۴	۱/۹۵
4.2.04	۲۵۸/۳۸	۸۷/۶۹۸۷	۳۱/۷۸۷۱	۳۰/۵۸	۳	۳/۷۹
4.2.05	۲۵۵/۵۱	۸۷/۰۷۶۷	۳۳/۰۲۱۵	۳۲/۰۳	۴	۳/۰۱
4.2.06	۲۵۸/۳۳	۸۳	۴۳/۸۹	۴۲/۶۶	۳	۲/۷۹
4.2.07	۲۵۸/۵۰	۸۷/۲۳۹۴	۳۲/۹۸	۳۱/۸۴	۲	۳/۴۴
5.1.09	۶۴/۸۳	۸۳/۲۹۶۲	۱۰/۸۲۹۱	۱۰/۵۶	۸	۲/۴۸
5.1.10	۶۴/۷۹	۷۳/۳۴۶۶	۱۷/۲۶	۱۶/۹۲	۲	۲/۱۹
5.1.11	۶۳/۸۱	۹۰/۶۶۴۵	۵/۹۵	۵/۸۲	۳	۱/۹۳
5.1.12	۶۳/۵۴	۸۷/۵۸۶۵	۷/۸۸	۷/۷۳	۱	۲/۲۵
5.1.13	۱۳/۳۰	۲۴/۴۷	۱۰/۰۵	۹/۸۲	۱	۱/۲۷
5.1.14	۶۴/۷۰	۷۸/۷۳	۱۳/۷۶	۱۳/۵۱	۴	۱/۸۱
5.2.08	۲۵۸/۸۶	۸۴/۶۷	۳۹/۵۲	۳۸/۲۳	۲	۳/۰۱
5.2.09	۲۵۷/۱۳	۷۸/۱۸	۵۶/۰۹	۵۴/۶۱	۴	۲/۶۳
5.2.10	۲۴۷/۸۳	۷۵/۲۰	۶۱/۴۴	۶۰/۲۶	۳	۱/۹۲
5.3.01	۱۰۰۲۴	۸۴/۹۹	۱۵۴/۰۶	۱۴۸/۴۹	۳	۳/۶۱
5.3.02	۱۰۰۳۳	۸۱/۴۰	۱۹۲/۱۹	۱۸۶/۳۷	۳	۳/۰۲
7.1.01	۲۵۷/۴۸	۸۴/۹۱	۳۸/۸۳	۳۷/۶۹	۱	۲/۹۳
7.1.02	۲۲۳/۸۳	۹۰/۹۴	۲۰/۲۶	۱۹/۵۵	۱	۳/۵۲
7.1.03	۲۵۸/۳۱	۸۴/۷۶	۳۹/۳۵	۳۸/۱۹	۱	۲/۹۴
7.1.04	۲۵۸/۰۸	۸۵/۴۸	۳۷/۴۵	۳۶/۳۳	۲	۲/۹
7.1.05	۲۵۸/۳۲	۸۰/۷۰	۴۹/۸۳	۴۸/۶۲	۱	۲/۴۲
7.1.06	۲۵۸/۵۲	۸۰/۷۵	۴۹/۷۶	۴۸/۵۰	۴	۲/۵۳
7.1.07	۲۵۸/۵۱	۸۱/۵۰	۴۷/۸۲	۴۶/۴۶	۴	۲/۸۴
7.1.08	۲۵۷/۵۰	۸۷/۵۲	۳۲/۱۱	۳۱/۰۸	۲	۳/۲۲
7.1.09	۲۵۸/۳۴	۸۲/۵۷	۴۵/۰۱	۴۴/۸۲	۴	۲/۶۲
7.1.10	۲۵۸/۴۹	۸۵/۲۵	۳۸/۱۲	۳۶/۹۶	۳	۳/۰۶
7.2.01	۱۰۰۳۳	۸۸/۷۶	۱۱۶/۰۴	۱۱۲/۰۸	۴	۳/۴۱
Boat	۲۵۸/۴۵	۸۴/۱۶	۴۰/۹۳	۳۹/۷۸	۲	۲/۸۰
Elaine	۲۵۷/۳۷	۸۶/۰۱	۳۵/۹۹	۳۴/۸۰	۳	۳/۳۱
gray21	۶/۴۴	۰/۰۳۹	۶/۴۴	۶/۴۰	۴	۰/۶
House	۷۷۵/۰۲	۹۳/۹۵	۴۰/۳۸	۳۹/۳۳	۳	۲/۶۰
Numbers	۲۳۱/۵۷	۷۲/۰۱	۶۴/۸۰	۶۳/۱۵	۳	۲/۵۴
Ruler	۵۵/۸۱	۲۰/۱۱	۴۴/۵۸	۴۱/۶۱	۱	۵/۳۳
Testpat	۱۸۸/۵۰	۵۸/۷۸	۷۷/۶۸	۷۵/۲۵	۴	۳/۱۳
						AVG(improve)=۲/۲۷۲



شکل ۴-۲۶: بهبود عملکرد JPEG با استفاده از روش پیشنهادی اول.

۴-۷-۵- نتایج پیش پردازش تصویر با روش پیشنهادی دوم در بهبود عملکرد Lossless JPEG

در جدول (۳-۴) حجم تصویر پردازش شده با احتساب سربار برای بازیابی تصویر درج شده است. حجم سربار برای کلیه تصاویر برابر با ۰/۰۸۶۹ کیلوبایت به دست آمده است و مقدار پارامتر مشخص کننده تعداد دسته‌ها برابر با دو می‌باشد. شکل (۴-۲۷)، نمودار مربوط به این جدول را نشان می‌دهد.



شکل ۴-۲۷: بهبود عملکرد lossless JPEG با استفاده از روش پیشنهادی دوم.

جدول ۴-۳: نتایج حاصل از اعمال روش پیشنهادی دوم بر روی تصاویر پایگاه داده USD.

All database	Original size (Kb)	PNSR of processed image	Size processed image in lossless JPEG(Kb)	Size in lossless JPEG (Kb)	CR lossless JPEG (%)	CR purposed (%)	Improve (%)
4.1.01	۶۴/۶۸	۴۶/۸۴	۳۱/۲۷	۳۸/۹۹	۳۹/۷۱	۵۱/۶۵	۱۱/۹۴
4.1.02	۶۰/۵۸	۴۶/۸۵	۳۰/۴۶	۳۸/۲۹	۳۶/۷۹	۴۹/۷۱	۱۲/۹۲
4.1.03	۶۳/۴۹	۴۶/۶۶	۲۲/۸۰	۳۰/۱۰	۵۲/۵۹	۶۴/۰۹	۱۱/۴۹
4.1.04	۶۴/۷۸	۴۶/۴۷	۳۱/۲۵	۳۹/۱۵	۳۹/۵۶	۵۱/۷۵	۱۲/۱۸
4.1.05	۶۴/۵۳	۴۶/۲۳	۲۸/۸۶	۳۶/۱۹	۴۳/۹۱	۵۵/۳۷	۱۱/۳۶
4.1.06	۶۳/۵۰	۴۶/۷۷	۳۷/۳۳	۴۵/۱۱	۲۸/۹۵	۴۱/۲۰	۱۲/۲۴
4.1.07	۵۳/۹۹	۴۵/۷۵	۱۹/۶۴	۲۵/۴۵	۵۲/۸۶	۶۳/۶۲	۱۰/۷۵
4.1.08	۵۷/۷۲	۴۵/۸۷	۲۳/۴۵	۳۰/۰۲	۴۶/۱۲	۵۷/۹۰	۱۱/۷۷
4.2.01	۲۵۵/۹۹	۴۵/۹۶	۱۰۹/۱۴	۱۳۷/۰۶	۴۶/۴۵	۵۷/۳۶	۱۰/۹۰
4.2.02	۲۵۶/۳۳	۴۶/۲۰	۱۲۵/۳۰	۱۵۷/۰۲	۳۸/۷۴	۵۱/۱۱	۱۲/۳۷
4.2.03	۲۵۸/۵۶	۴۶/۲۴	۱۷۵/۸۲	۲۰۷/۱۰	۱۹/۹۰	۳۲	۱۲/۰۹
4.2.04	۲۵۸/۳۸	۴۵/۹۴	۱۳۴/۲۵	۱۶۵/۱۴	۳۶/۰۸۶	۴۸/۰۳	۱۱/۹۵
4.2.05	۲۵۵/۵۱	۴۶/۱۹	۱۱۹/۸۳	۱۵۰/۹۲	۴۰/۹۳	۵۳/۱۰	۱۲/۱۶
4.2.06	۲۵۸/۳۳	۴۶/۶۸	۱۵۱/۳۲	۱۸۲/۶۳	۲۹/۳۰	۴۱/۴۲	۱۲/۱۱
4.2.07	۲۵۸/۵۰	۴۶/۶۰	۱۳۴/۸۳	۱۶۶/۱۷	۳۵/۷۱۶	۴۷/۸۴۱	۱۲/۱۲
5.1.09	۶۴/۸۳	۴۶/۲۱	۳۷/۳۲	۴۵/۱۰	۳۰/۴۱	۴۲/۴۳	۱۲/۰۱
5.1.10	۶۴/۷۹	۴۷/۵۷	۴۲/۶۳	۵۰/۳۱	۲۲/۳۳	۳۴/۱۹۵	۱۱/۸۵
5.1.11	۶۳/۸۱	۴۶/۹۲	۲۴/۹۷	۳۲/۱۴	۴۹/۶۲	۶۰/۸۶	۱۱/۲۳
5.1.12	۶۳/۵۴	۴۶/۸۴	۳۰/۰۳	۳۷/۳۶	۴۱/۱۹	۵۲/۷۳	۱۱/۵۴
5.1.13	۱۳/۳۰	۵۵/۹۳	۱۲/۱۰	۱۰/۴۲	۹/۰۱	۲۱/۶۴	۱۲/۶۲
5.1.14	۶۴/۷۰۷۰	۴۶/۴۹۰۱	۳۷/۹۲۷۷	۴۵/۷۷	۲۹/۲۶	۴۱/۳۸	۱۲/۱۲۰
5.2.08	۲۵۸/۸۶	۴۵/۱۱	۱۳۳/۵۶	۱۶۵/۰۸	۳۵/۹۷	۴۸/۲۰	۱۲/۲۲
5.2.09	۲۵۷/۱۳	۴۶/۶۱	۱۵۲/۷۱	۱۸۴/۶۲	۲۸/۱۹	۴۰/۶۰	۱۲/۴۰

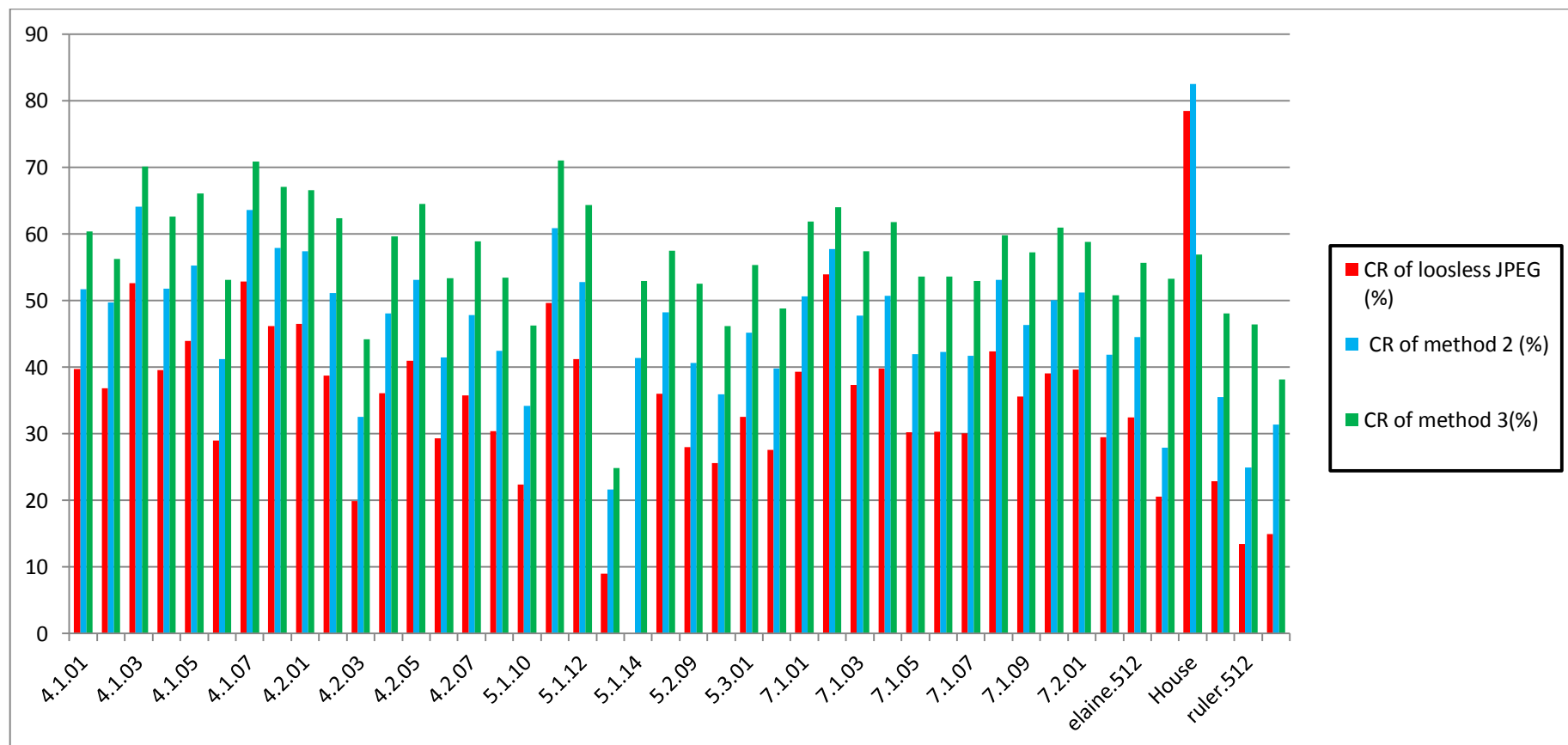
5.2.10	୨୪୮/୮୩	୪୮/୧୩	୧୫୮/୮୫	୧୮୪/୩୨	୨୫/୫୨	୩୫/୨୦	୧୦/୩୦
5.3.01	୧୦୨୪	୪୫/୮୮	୫୧୨/୮୧	୧୨୨/୮୨	୩୨/୫୦	୪୫/୧୮	୧୨/୧୮
5.3.02	୧୦୩୩	୪୫/୧୫	୧୨୨/୩୩	୮୪୮/୫୧	୨୮/୫୮	୩୨/୮୮	୧୨/୨୦
7.1.01	୨୫୮/୪୮	୪୧/୮୧	୧୨୮/୨୨	୧୫୧/୨୮	୩୨/୩୦	୫୦/୫୮	୧୧/୨୮
7.1.02	୨୨୩/୮୩	୪୮	୨୪/୧୦	୧୦୩/୦୪	୫୩/୨୧	୫୮/୮୩	୩/୮୮
7.1.03	୨୫୮/୩୧	୪୮/୩୮	୧୩୫/୦୨	୧୧୮/୧୮	୩୮/୩୧	୪୮/୮୨	୧୦/୪୧
7.1.04	୨୫୮/୦୮	୪୧/୪୨	୧୨୮/୧୨	୧୫୫/୩୫	୩୨/୮୦	୫୦/୮୧	୧୦/୨୦
7.1.05	୨୫୮/୩୨	୪୧/୧୧	୧୪୨/୨୮	୧୮୦/୨୩	୩୦/୨୩	୪୧/୨୪	୧୧/୮୧
7.1.06	୨୫୮/୫୨	୪୧/୪୫	୧୪୨/୨୦	୧୮୦/୦୮	୩୦/୩୪	୪୨/୨୮	୧୧/୨୪
7.1.07	୨୫୮/୫୧	୪୧/୩୪	୧୫୦/୧୩	୧୮୦/୮୫	୩୦/୦୮	୪୧/୮୩	୧୧/୧୫
7.1.08	୨୫୮/୫୦	୪୧/୨୨	୧୨୦/୮୮	୧୪୮/୫୨	୪୨/୩୨	୫୩/୦୨	୧୦/୮୮
7.1.09	୨୫୮/୩୪	୪୮	୧୩୮/୮୦	୧୧୧/୪୧	୩୫/୫୧	୪୧/୩୦	୧୦/୮୪
7.1.10	୨୫୮/୪୨	୪୧/୩୫	୧୨୨/୨୪	୧୫୮/୪୮	୩୨/୦୮	୫୦	୧୦/୨୨
7.2.01	୧୦୩୩	୪୮/୨୪	୫୦୩/୨୮	୧୨୩/୧୨	୩୨/୧୨୨	୫୧/୨୨	୧୧/୫୮
boat.512	୨୫୮/୪୫	୪୧/୦୫	୧୫୦/୩୪	୧୮୨/୨୩	୨୨/୪୨	୪୧/୮୩	୧୨/୩୩
elaine.512	୨୫୮/୩୮	୪୫/୫୮୦୪	୧୪୨/୮୪	୧୮୩/୮୩	୩୨/୪୫	୪୪/୫୩	୧୨/୦୮
gray21.512	୧/୪୪	୪୮/୩୪	୫/୧୧	୪/୧୪	୨୦/୫୫	୨୮/୨୪	୮/୩୨
House	୮୮୫/୦୨	୪୧/୫୨	୧୩୫/୧୮	୧୧୮/୨୫	୮୮/୪୧	୮୨/୪୨	୪/୦୮
numbers.512	୨୩୧/୫୮	୪୫/୮୮	୧୪୨/୩୧	୧୮୮/୧୫	୨୨/୮୫	୩୫/୫୨	୧୨/୧୧
ruler.512	୫୫/୮୧	୫୮/୮୧	୪୮/୨୨	୪୧/୨୧	୧୩/୪୧	୨୪/୨	୧୧/୪୪
testpat.1k	୧୮୮/୫୦	୪୮/୩୧	୧୧୦/୩୨	୧୨୨/୪୨	୧୪/୨୫	୩୧/୩୪	୧୧/୩୨
					AVG: ୩୪/୫୩	AVG: ୪୫/୧୨	AVG: ୧୧/୧୨

روش پیشنهادی دوم قادر است به‌طور میانگین، نرخ فشرده‌سازی را به میزان ۱۱/۱۲ درصد برای تصاویر پایگاه داده USD افزایش دهد و در عین حال اطلاعات ناچیزی از تصویر را از دست می‌دهد.

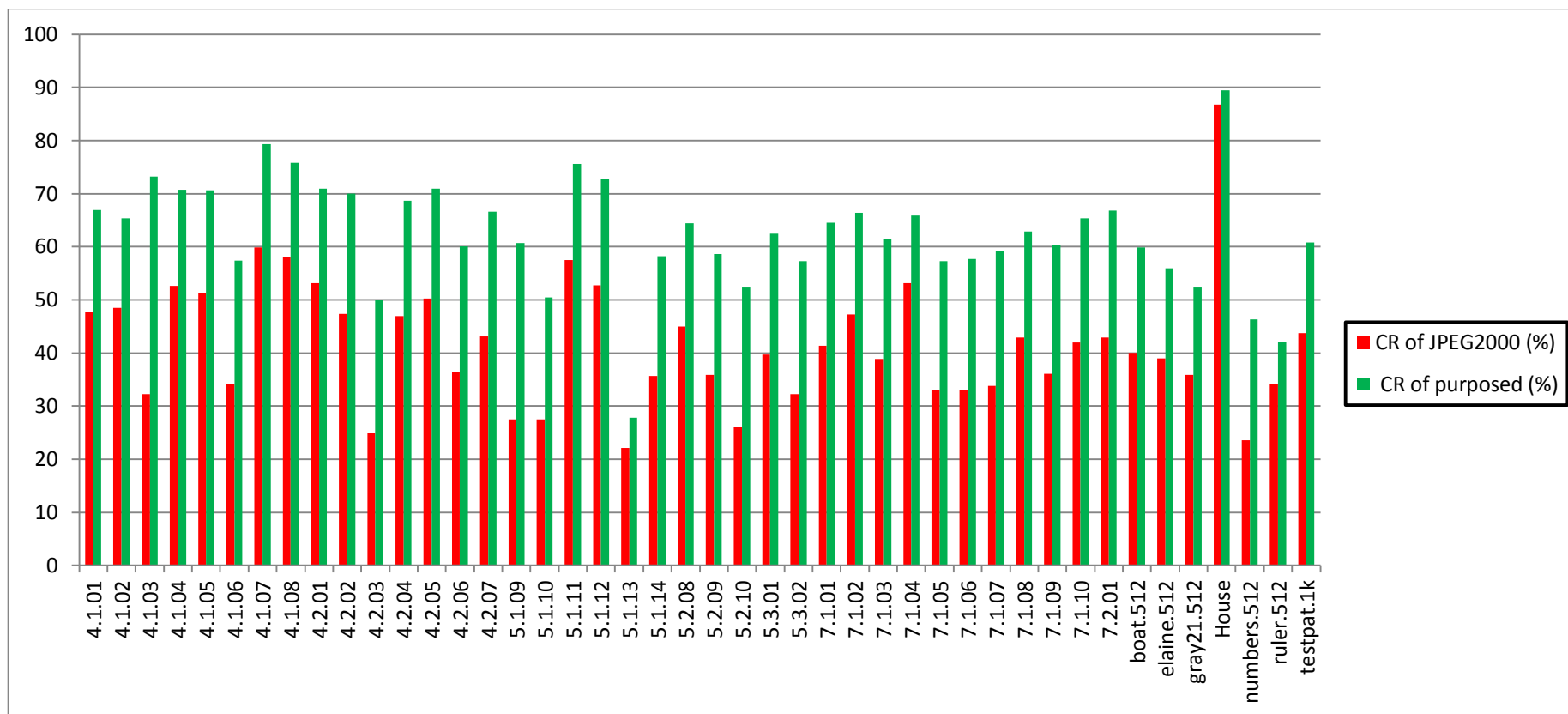
۴-۷-۶- نتایج پیش‌پردازش تصویر با روش پیشنهادی سوم در بهبود عملکرد Lossless JPEG و JPEG2000

روش پیشنهادی سوم، بر روی تمام تصاویر پایگاه داده USD مورد آزمایش و ارزیابی قرار گرفته است. با توجه به نتایج به دست آمده در اغلب موارد بلوک‌بندی تصویر حتی با ابعاد ۸×۸ نیز باعث بهبود در نرخ فشرده‌سازی می‌شود. اما حجم قابل توجهی باید به عنوان سربار لازم برای بازیابی ذخیره گردد. با افزایش ابعاد بلوک‌بندی، حجم سربار کاهش یافته و نرخ فشرده‌سازی افزایش می‌یابد. اما کیفیت تصویر پردازش شده کم می‌شود. به همین دلیل ابعاد مناسب برای فشرده‌سازی بر اساس حجم سربار، میزان بهبود در نرخ فشرده‌سازی و میزان کیفیت تصویر انتخاب می‌گردد. نتایج نشان می‌دهد که معمولاً ابعاد بزرگتر از ۳۲×۳۲ مناسب‌تر خواهد بود. با توجه به نتایج به دست آمده میانگین درصد بهبود بر اساس مناسب‌ترین ابعاد بلوک‌بندی، برای کلیه تصاویر پایگاه داده USD محاسبه شده است. نتایج نشان می‌دهد که نرخ فشرده‌سازی Lossless JPEG با استفاده از این پیش‌پردازش به طور میانگین به میزان ۲۱/۳۷ درصد و نرخ فشرده‌سازی JPEG2000 به طور میانگین ۲۳/۳۳ درصد بهبود یافته است. جداول مربوط به نتایج به دست آمده از روش پیشنهادی سوم، در پیوست (۲) آمده است.

شکل (۴-۲۸)، مقایسه‌ای بین عملکرد الگوریتم فشرده‌سازی Lossless JPEG، روش پیشنهادی دوم و روش پیشنهادی سوم را نشان می‌دهد. در شکل (۴-۲۹) نمودار مربوط به نرخ فشرده‌سازی الگوریتم JPEG2000 و نرخ فشرده‌سازی روش پیشنهادی سوم، برای تصاویر پایگاه داده USD نشان داده شده است.



شکل ۴-۲۸: مقایسه عملکرد lossless JPEG با روش پیشنهادی دوم و سوم بر روی تصاویر پایگاه داده USD.



شکل ۴-۲۹: بهبود عملکرد JPEG2000 با استفاده از روش پیشنهادی سوم.

۴-۷-۷- نتایج روش پیشنهادی چهارم در بهبود عملکرد الگوریتم CIC

جدول ۴-۴: نتایج روش پیشنهادی چهارم بر روی تصاویر پایگاه داده Kodak.

All database	Original size(Kb)	Size compressed image (Kb)		CR(%)		Improve (%)
		With method 4	Without method 4	With method 4	Without method 4	
Kodim01	۷۱۹	۷۰/۴	۷۴/۶	۹۰/۲۰	۸۹/۶۲	۰/۵۸
Kodim02	۶۰۳	۶۸/۹	۷۱/۹	۸۸/۵۷	۸۸/۰۷	۰/۵
Kodim03	۴۹۱	۵۱/۱	۵۳/۳	۸۹/۵۹	۸۹/۱۴	۰/۴۵
Kodim04	۶۲۲	۶۱/۸	۶۳/۳	۹۰/۰۶	۸۹/۸۲	۰/۲۴
Kodim05	۷۶۷	۶۷/۶	۶۹/۱	۹۱/۱۸	۹۰/۹۹	۰/۱۹
Kodim06	۶۰۴	۵۸/۹	۶۰/۲	۹۰/۲۴	۹۰/۰۳	۰/۲۱
Kodim07	۵۵۳	۶۱/۱	۶۳/۲	۸۸/۹۵	۸۸/۵۵	۰/۴۰
Kodim08	۷۶۹	۶۹/۴	۷۲/۷	۹۰/۹۷	۹۰/۵۴	۰/۴۳
Kodim09	۵۶۹	۳۵/۷۶	۳۷	۹۳/۷۱	۹۳/۴۹	۰/۲۲
Kodim10	۵۷۹	۴۲/۸	۴۴/۷	۹۲/۶۰	۹۲/۲۷	۰/۳۳
Kodim11	۶۰۶	۶۳/۹	۶۵/۱	۸۹/۴۵	۸۹/۲۵	۰/۲۰
Kodim12	۵۱۸	۴۲/۱	۴۴/۶	۹۱/۸۷	۹۱/۳۸	۰/۴۹
Kodim13	۸۰۳	۷۶/۹	۷۷/۷	۹۰/۴۲	۹۰/۳۲	۰/۱۰
Kodim14	۶۷۵	۵۴/۱	۵۷/۲	۹۱/۹۸	۹۱/۵۲	۰/۴۶
Kodim15	۵۹۸	۵۳/۹	۵۶	۹۰/۹۸	۹۰/۶۳	۰/۳۵
Kodim16	۵۲۱	۷۱/۵۹	۷۳/۸	۸۶/۲۵	۸۵/۸۳	۰/۴۲
Kodim17	۵۸۷	۵۴/۶	۵۷/۱	۹۰/۶۹	۹۰/۲۷	۰/۴۲
Kodim18	۷۶۲	۶۳/۸	۶۵/۹	۹۱/۶۲	۹۱/۳۵	۰/۲۷
Kodim19	۶۵۵	۴۹/۵	۵۰/۹	۹۲/۴۴	۹۲/۲۲	۰/۲۲
Kodim20	۴۸۰	۴۱/۴	۴۲/۸	۹۱/۳۷	۹۱/۰۸	۰/۲۹
Kodim21	۶۲۲	۵۸/۵	۶۰/۷	۹۰/۵۹	۹۰/۲۴	۰/۳۵
Kodim22	۶۸۵	۵۰/۸	۵۲/۷	۹۲/۵۸	۹۲/۳۰	۰/۲۸
Kodim23	۵۴۴	۴۱/۲۹	۴۳/۶	۹۲/۴۰	۹۱/۹۸	۰/۴۲
Kodim24	۶۸۹	۶۳/۷	۶۵/۲	۹۰/۷۵	۹۰/۵۳	۰/۲۲

الگوریتم CIC قادر است یک تصویر رنگی را با نرخ بسیار بالا فشرده کند. با این حال همان‌طور که در

جدول (۴-۴) قابل مشاهده است، تغییر فضای رنگ تصویر نیز باعث بهبود عملکرد آن می‌شود.

۴-۷-۸- بررسی زمان پردازش روش‌های پیشنهادی

در روش‌های پیشنهادی اول و دوم، با افزایش ابعاد تصویر زمان پردازش، افزایش می‌یابد. در روش

پیشنهادی سوم، زمان پردازش به ابعاد تصویر اولیه و ابعاد بلوک‌بندی تصویر وابسته است. هر چه ابعاد

تصویر اولیه کوچکتر باشد، زمان پردازش کمتر است و هر چه ابعاد بلوک‌بندی تصویر کوچکتر باشد، زمان پردازش بیشتر خواهد بود. با توجه به نتایج به‌دست آمده، میانگین زمان محاسباتی برای روش اول ۶/۴۳ ثانیه و برای روش دوم ۴/۵۹ ثانیه به دست آمده است. همچنین میانگین زمان محاسباتی برای روش پیشنهادی سوم بر اساس بهترین سایز بلوک‌بندی، برای بهبود JPEG2000 و LosslessJPEG به ترتیب برابر با ۳/۱۹ و ۳/۲۶ ثانیه است. زمان اعمال هر یک از روش‌های پیشنهادی، بر روی تصاویر پایگاه داده USD در پیوست (۲) نشان داده شده است.

۴-۸- مقایسه کارایی روش‌های پیشنهادی با روش‌های موجود

زمانی مقایسه میان چند روش، معنادار خواهد بود که اولاً پایگاه داده و تصاویری که مورد آزمایش قرار می‌گیرند، یکسان باشند. ثانیاً ارزیابی با معیارهای یکسان صورت گرفته باشد. ثالثاً، الگوریتم فشرده‌سازی که یک پیش‌پردازش خاص منجر به بهبود آن شده است، یکسان باشد. لذا با توجه به این موارد امکان مقایسه نتایج با تمام مقالاتی که در فصل سوم مطرح شدند، فراهم نبود. اما به منظور انجام مقایسه، نتایج اعمال روش پیشنهادی خود را با روش مطرح شده در [۵۲] مورد مقایسه قرار می‌دهیم (جدول (۴-۵)).

جدول ۴-۵: مقایسه بین روش پیشنهادی سوم با BWCA در بهبود عملکرد JPEG.

Image	Lossless JPEG	JPEG2000	BWCA	Method3+ Lossless JPEG	Method3+ JPEG2000
Hand primary	۲/۰۱۹	۲/۵۵۷	۲/۱۲۳	۳/۱۴۵	۲/۴۲۲
Hand secondary	۲/۰۳۷	۳/۱۳۴	۲/۶۹۳	۳/۴۱	۲/۴۴۵
Pelvis primary	۲/۱۸۸	۲/۰۱۹	۱/۷۹۵	۲/۴۸۳	۲/۶۲۵
Thoraces frontal primary	۲/۰۴۸	۲/۳۸۳	۲/۰۳۵	۲/۹۳۱	۲/۴۵۷
Thoraces frontal secondary	۲/۶۱۲	۳/۳۹۴	۲/۹۸۲	۳/۶۴۳	۳/۱۳۴
Thoraces lateral primary	۲/۲۹۶	۲/۸۲۲	۲/۳۷۹	۳/۳۶۹	۲/۷۵۵
Thoraces frontal secondary	۲/۵۲۸	۳/۱۷۱	۲/۷۵۳	۳/۸۰۴	۳/۰۳۳
AVG CR	۲/۲۴۶	۲/۷۸۲	۲/۳۹۴	۳/۲۵۵	۲/۶۹۵

۴-۹- نتیجه‌گیری

در این فصل، چهار روش پیشنهادی معرفی شد. سعی نمودیم تا حد امکان روش‌ها ساده، با زمان پردازش کم و بازگشت‌پذیر باشند. نتایج نشان می‌دهند که پیش‌پردازش‌های مطرح شده، توانسته‌اند عملکرد الگوریتم‌های فشرده‌سازی را بهبود داده و در عین حال کیفیت تصویر را حفظ نمایند.

فصل پنجم

نتیجه‌گیری و پیشنهادات برای ادامه کار

۵-۱- نتیجه گیری

در این پایان نامه در فصل اول به مفاهیم فشرده سازی، اهمیت و ضرورت پیش پردازش در فشرده سازی پرداختیم. روند رو به رشد اطلاعات دیجیتال از یک سو و نیاز به ذخیره ی این اطلاعات از سوی دیگر باعث شده که فشرده سازی به عنوان یک نیاز ضروری مطرح شود و روش های زیادی برای فشرده نمودن اطلاعات ایجاد گردد. در فصل دوم، تعدادی از این روش های فشرده سازی را معرفی نمودیم.

الگوریتم های فشرده سازی سعی می کنند با در نظر گرفتن یک یا چند ویژگی در تصویر، افزونگی را شناسایی نموده و با حذف افزونگی در تصویر، حجم آن را کاهش دهند. هر یک از الگوریتم های فشرده سازی تصویر، به ازای تصاویر مختلف، نرخ فشرده سازی یکسانی را ارائه نمی دهند. نرخ های فشرده سازی این الگوریتم ها، ممکن است به عواملی چون تنوع سطوح خاکستری، هموار بودن تصویر و یکسان بودن مقادیر پیکسل های مجاور در تصویر، بستگی داشته باشد. بر این اساس، انجام یک پیش پردازش مناسب بر روی تصویر می تواند در تقویت پارامترهای تاثیرگذار در فشرده سازی موثر باشد. بر این اساس، تاکنون محققان روش های پیش پردازشی را برای بهبود عملکرد الگوریتم های فشرده سازی معرفی نموده اند. در فصل سوم، روش های موجود در این حوزه را معرفی نمودیم. در فصل چهارم، چهار روش پیش پردازش جدید را به منظور بهبود عملکرد چند الگوریتم فشرده سازی پرکاربرد ارائه نموده و مورد ارزیابی قرار دادیم.

پیش پردازش اول بدون اتلاف بوده و در دو گام انجام شده است. در گام اول، تصویر به بلوک های 8×8 بدون هم پوشانی تقسیم شده و بلوک ها بر اساس افزایش (کاهش) ضرایب DC، مرتب شده اند. در گام دوم، هشت تبدیل هندسی برگشت پذیر بر روی تصویر حاصل از گام اول اعمال شده و در نهایت از بین تصاویر حاصل، بهترین تصویر برای ورود به مرحله ی فشرده سازی انتخاب شده است. این پردازش باعث کاهش دامنه ی تغییرات، در نتایج حاصل از محاسبات JPEG شده و نرخ فشرده سازی را بهبود داده است.

در روش دوم، با انجام یک پیش پردازش، تنوع سطوح خاکستری کاهش می یابد. سطوح خاکستری که از نظر مقدار، به هم نزدیک می باشند، در یک دسته قرار می گیرند. سپس، در هر دسته، یک سطح خاکستری به عنوان نماینده انتخاب شده و مقدار سطح خاکستری هر پیکسل از تصویر اصلی با مقدار نماینده ی مربوط به دسته ی خودش، جایگزین می گردد. در این روش، دسته بندی بر اساس یک پارامتر صورت گرفته است. هر چه مقدار این پارامتر بزرگتر باشد، سطوح خاکستری بیشتری در یک دسته قرار گرفته و در نتیجه کاهش تنوع سطوح خاکستری بیشتر می شود، اما کیفیت تصویر پردازش شده کاهش می یابد، به همین دلیل این مقدار به نحوی انتخاب شد که خرابی تصویر حداقل باشد. از طرف دیگر برای کاهش میزان خرابی تصویر، در هر دسته، سطحی به عنوان نماینده انتخاب گردید که در بین سایر سطوح آن دسته بیشترین فراوانی را در تصویر اصلی داشته باشد. پس از انجام این مراحل، در آخرین مرحله هیستوگرام تصویر متراکم شد. این پیش پردازش موجب کاهش آنتروپی و افزایش افزونگی شده و با حفظ کیفیت تصویر، توانست نرخ فشرده سازی الگوریتم Lossless JPEG را بهبود دهد.

در روش سوم، ابتدا تصویر در ابعاد ثابت بلوک بندی شده و پردازش معرفی شده در روش سوم، به طور مجزا برای هر بلوک انجام شد. با این تفاوت که، مقدار پارامتر تعیین کننده ی تعداد سطوح هر دسته، برای هر بلوک متفاوت بود. برای نواحی که چشم انسان نسبت به تغییرات حساس است مقدار کوچکتر و برای نواحی که چشم انسان حساسیت کمتری نسبت به تغییرات دارد، مقدار بزرگتری انتخاب شد. برای هر تصویر، پردازش با ابعاد بلوک بندی مختلف انجام شده و بهترین بلوک بندی (بر اساس میزان سربار و نرخ فشرده سازی) برای هر تصویر انتخاب گردید. با این پردازش آنتروپی کاهش یافته و افزونگی افزایش می یابد، به همین علت، تصویر پردازش شده می تواند با نرخ بیشتری فشرده شود. این پیش پردازش باعث بهبود نرخ فشرده سازی الگوریتم های JPEG2000 و Lossless JPEG شده است.

در روش چهارم نیز، با استفاده از الگوریتم ژنتیک، فضای رنگ مناسبی برای فشردسازی هر تصویر به طور مجزا ایجاد شد. فضای رنگ به گونه‌ای است که اکثر انرژی در یک کانال از تصویر تجمع می‌یابد و در عین حال وابستگی بین هر دو کانال از تصویر تا حد امکان حداقل است. این پیش‌پردازش بر روی الگوریتم فشردسازی با نام CIC مورد ارزیابی قرار گرفته و منجر به بهبود عملکرد آن شده است.

هر یک از پیش‌پردازش‌های معرفی شده، منحصر با توجه به عملکرد یک الگوریتم فشردسازی خاص ارائه شدند. تمامی روش‌ها توانستند عملکرد الگوریتم‌های فشردسازی را بهبود داده و در عین حال کیفیت تصویر پردازش شده را حفظ نمایند.

۵-۲- پیشنهادات برای ادامه‌ی کار

الگوریتم‌های بسیار متنوعی برای فشردسازی تصویر وجود دارند. هر یک از این الگوریتم‌ها عملکرد منحصر به فرد خود را داشته و غالباً هر پیش‌پردازش، می‌تواند به بهبود عملکرد یک الگوریتم فشرد-سازی کمک کند. این امر باعث گستردگی موضوع مطرح شده در این پژوهش شده است. در واقع به تعداد الگوریتم‌های فشردسازی می‌توان پیش‌پردازش‌هایی برای بهبود عملکرد آنها پیشنهاد داد. در ادامه به چند موضوع برای پژوهش‌های آتی اشاره می‌شود:

- تقسیم تصویر به پنجره‌هایی با ابعاد متغیر و یافتن راهی برای ذخیره‌ی اطلاعات لازم برای بازیابی، در روش پیشنهادی سوم، راهکار دیگری است که می‌تواند به عنوان پژوهش‌های آتی در نظر گرفته شود.
- تقسیم تصویر به طور سلسله‌مراتبی نیز می‌تواند در بهبود روش پیشنهادی سوم موثر باشد. با این روش لازم است که تصویر در ابتدا به چند بخش مجزا و بدون هم‌پوشانی تقسیم شده و هر بخش به طور جداگانه پردازش شود. اگر خرابی بخش پردازش شده بیش از حد مجاز شد، دوباره این بخش به چند زیر بخش تقسیم شده و پردازش برای هر قسمت به طور جداگانه صورت گیرد.
- می‌توان چند روش پیش‌پردازش را با هم ترکیب نموده و سپس تصویر حاصل را به عنوان ورودی الگوریتم فشردسازی در نظر گرفت. البته در هر مرحله از پیش‌پردازش، باید در نظر

داشت که پیش‌پردازش فعلی منجر به از دست رفتن اطلاعات لازم برای بازیابی تصویر پردازش شده‌ی مرحله‌ی قبل نشود.

- در الگوریتم‌های فشرده‌سازی که تصویر را از فضای رنگ RGB به فضای رنگ YCBCR منتقل می‌کنند، استفاده و اعمال روش پیشنهادی چهارم می‌تواند نتایج فشرده‌سازی را بهبود دهد. به عنوان مثال عملکرد الگوریتم‌های JPEG و JPEG2000 با این روش، قطعا بهبود خواهد یافت.

پیوست (۱)

پ-۱-۱- الگوریتم ژنتیک

ژنتیک یک الگوریتم تکاملی است. این الگوریتم یک تکنیک جستجو برای یافتن راه‌حل‌های بهینه در مسائل جستجو است. مجموعه‌ی اولیه، یک مجموعه‌ی کاملاً تصادفی از موجودیت‌هاست و در نسل‌های بعد، هر بار مناسب‌ترین‌ها و نه الزاماً بهترین‌ها، انتخاب می‌شوند.

هر راه‌حل، یک لیستی از پارامترها است که ژنوم یا کروموزوم نامیده می‌شوند. در ابتدا چندین کروموزوم، به عنوان نسل اولیه تولید می‌شوند. در طول هر نسل هر یک از آنها با استفاده از تابع برازش مورد ارزیابی قرار می‌گیرند. سپس ایجاد دومین نسل از جامعه با استفاده از فرایندهای انتخاب، ترکیب، جهش و تولید، آغاز می‌گردد. انتخاب طوری انجام می‌شود که حتی ضعیف‌ترین عناصر هم شانس برای انتخاب شدن داشته باشند. بنابراین از نزدیک شدن به جواب محلی جلوگیری می‌شود. در ادامه به توضیح مفصل‌تر این الگوریتم خواهیم پرداخت [۶۷-۶۸].

پ-۱-۱-۱- ساختار الگوریتم ژنتیک

به‌طور کلی الگوریتم‌های ژنتیک، از اجزای زیر تشکیل می‌شوند.

کروموزوم: هر کروموزوم نشان‌دهنده‌ی یک نقطه در فضای جستجو و یک راه‌حل ممکن برای مسئله است. هر کروموزوم از تعدادی ژن (متغیر) تشکیل می‌شود.

جمعیت: مجموعه‌ای از کروموزوم‌ها در کنار هم یک جمعیت را تشکیل می‌دهند. با تاثیر عملگرهای ژنتیکی بر روی هر جمعیت، جمعیت جدیدی با همان تعداد کروموزوم ساخته می‌شود.

تابع برازندگی: به‌منظور حل هر مسئله با استفاده از الگوریتم‌های ژنتیکی، باید یک تابع برازندگی برای مسئله ابداع شود، برای هر کروموزوم، این تابع، عددی غیرمنفی را باز می‌گرداند.

پ-۱-۱-۲- عملگرهای الگوریتم ژنتیک

در طی مرحله‌ی تولیدمثل از عملگرهای ژنتیکی استفاده می‌شود. عملگر انتخاب، ترکیب و جهش بیشترین کاربرد را در الگوریتم ژنتیک دارند.

عملگر انتخاب: این عملگر از بین کروموزوم‌های موجود تعدادی کروموزوم را برای تولید مثل انتخاب می‌کند. کروموزوم‌های شایسته‌تر شانس انتخاب بیشتری دارند. عملگر انتخاب تعیین می‌کند که جمعیت چگونه از یک نسل به نسل دیگر به‌روز رسانی شود.

عملگر ترکیب: این عملگر، صفات ژنتیکی افراد را برای تولید فرزند به گونه‌ای ترکیب می‌کند که برخی از ویژگی‌های والدین حفظ شود. هدف این عملگر تولید فرزندانی است که نسبت به اجداد خود شایستگی بالاتری دارند. نتیجه‌ی این عملگر، همگرایی سریع‌تر است.

عملگر جهش: این عملگر به طور تصادفی یک ژن از یک فرزند را پس از ترکیب تغییر می‌دهد. اگر ژن‌ها باینری باشند، جهش، ژن را به وارونش تبدیل می‌کند و اگر متعلق به یک مجموعه باشد، عنصر دیگری از آن مجموعه را به جای آن قرار می‌دهد. جهش، در واقع یک عملگر حمایتی برای بازیابی ویژگی‌های ژنتیکی از دست‌رفته است. در الگوریتم ژنتیک، احتمال جهش بین $0/01$ تا $0/001$ در نظر گرفته می‌شود. می‌توان امیدوار بود که به کمک این عملگر، کروموزوم‌های خوبی که در مرحله‌ی انتخاب یا تکثیر حذف شده‌اند دوباره احیا گردند. با وجود این الگوریتم می‌توان تضمین نمود که بدون توجه به پراکندگی جمعیت اولیه، احتمال جستجوی هر نقطه از فضای مسئله هیچگاه صفر نیست.

پ-۱-۱-۳- شرط اتمام الگوریتم ژنتیک

موارد زیر می‌تواند به عنوان شروط اتمام الگوریتم، مورد استفاده قرار گیرند:

- رسیدن به تعداد ثابتی از نسل‌ها
- اتمام زمان / بودجه
- عدم تغییر بهترین شایستگی در یک محدوده‌ی زمانی مشخص
- عدم بهبود در بهترین شایستگی در طی چند نسل متوالی
- پیدا شدن فرزندی که بهترین ملاک را برآورده کند

پ-۱-۱-۴- مزایا و نقاط قوت الگوریتم ژنتیک

- موازی بودن الگوریتم ژنتیک یکی از مهمترین نقاط قوت آن محسوب می‌شود. چون ژنتیک چندین نقطه‌ی آغاز دارد، در یک لحظه، قادر است فضای مسئله را در چند جهت جستجو کند و اگر یکی به جواب نرسید سایر راه‌ها را ادامه دهد.
- الگوریتم‌های ژنتیک هیچ اطلاعی در مورد مسائلی که حل می‌کنند ندارند. چون تنها تغییراتی تصادفی در راه‌حل‌های کاندید ایجاد می‌کنند و سپس با تابع برازش بررسی می‌کنند که تغییرات باعث پیشرفت شده‌اند یا خیر. بنابراین در این الگوریتم تمام راه‌حل‌های ممکن به روی مسئله باز است. اما در روش‌هایی که با اطلاعات محدود می‌شوند، از راه قیاس تصمیم گرفته می‌شود در نتیجه راه‌حل‌های نو و جدید از بین می‌روند.
- الگوریتم‌های ژنتیک دارای قدرت بسیار زیادی در یافتن جواب مسئله می‌باشند، شاید بتوان کاربرد اصلی این الگوریتم‌ها را در مسائلی در نظر گرفت که دارای فضای حالت بسیار بزرگ هستند و عملاً بررسی همه حالت‌ها برای انسان در زمان‌های نرمال ممکن نیست.

پیوست (۲)

در این پیوست، نتایج حاصل از اعمال روش پیشنهادی سوم (دسته‌بندی سطوح خاکستری و متراکم کردن هیستوگرام بر اساس سیستم بینایی انسان) و زمان اعمال هر یک از روش‌های پیشنهادی، بر روی تصاویر پایگاه داده USD نشان داده شده است.

جدول پ-۲-۱: نتایج حاصل از اعمال روش پیشنهادی سوم در بهبود Lossless JPEG بر روی تصاویر پایگاه داده USD.

All database	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
4.1.01	۸	۶۴/۶۸	۳۹/۷۱	۳۴/۲۰	۴۷/۱۱	۷/۴۰	۰/۹۶۸۲	۶/۱۰۶۴	۰/۶۴۰۶
	۱۶	۶۴/۶۸	۳۹/۷۱	۲۹/۲۶	۵۴/۷۴	۱۵/۰۳	۰/۹۶۵۱	۲/۴۱۰۲	۰/۲۷۱۵
	۳۲	۶۴/۶۸	۳۹/۷۱	۲۷/۲۱	۵۷/۹۳	۱۸/۲۲	۰/۹۶۱۸	۰/۹۵۵۱	۰/۱۴۹۴
	۶۴	۶۴/۶۸	۳۹/۷۱	۲۶/۲۲	۵۹/۴۶	۱۹/۹۶	۰/۹۶۰۸	۰/۳۷۶۰	۰/۱۰۲۵
	۱۲۸	۶۴/۶۸	۳۹/۷۱	۲۵/۶۲	۶۰/۳۸	۲۰/۶۷	۰/۹۶۲۲	۰/۱۹۵۳	۰/۰۸۷۹
	۲۵۶	۶۴/۶۸	۳۹/۷۱	۲۷/۳۴	۵۷/۷۳	۱۸/۰۲	۰/۹۶۹۵	۰/۱۲۰۱	۰/۰۸۴
4.1.02	۸	۶۰/۵۸	۳۶/۷۹	۳۲/۶	۴۶/۱۳	۹/۳۳	۰/۹۵۸۷	۵/۶۹۴۳	۰/۶۳۲۸
	۱۶	۶۰/۵۸	۳۶/۷۹	۲۸/۳۴	۵۳/۲۱	۱۶/۴۲	۰/۹۴۷۶	۲/۲۴۸	۰/۲۶۹۵
	۳۲	۶۰/۵۸	۳۶/۷۹	۲۶/۶۷	۵۵/۹۶	۱۹/۱۷	۰/۹۳۶۶	۰/۹۳۷۵	۰/۱۴۹۴
	۶۴	۶۰/۵۸	۳۶/۷۹	۲۶/۳۴	۵۶/۵۱	۱۹/۷۲	۰/۹۳۵۸	۰/۳۷۱۱	۰/۱۰۲۵
	۱۲۸	۶۰/۵۸	۳۶/۷۹	۲۵/۸۹	۵۶/۲۲	۱۹/۴۲	۰/۹۳۱۵	۰/۱۸۱۶	۰/۰۸۷۹
	۲۵۶	۶۰/۵۸	۳۶/۷۹	۲۶/۵۲	۵۷/۲۶	۲۰/۴۷	۰/۹۳۰۴	۰/۱۲۱۱	۰/۰۸۴
4.1.03	۸	۶۳/۴۹	۵۲/۵۹	۲۷/۰۶	۵۷/۳۷	۴/۷۷	۰/۸۹۰۱	۴/۰۰۳۹	۰/۵۴۸۸
	۱۶	۶۳/۴۹	۵۲/۵۹	۲۳/۲۲	۶۳/۴۲	۱۰/۸۲	۰/۸۷۵۵	۱/۶۸۳۶	۰/۲۵
	۳۲	۶۳/۴۹	۵۲/۵۹	۲۱/۳۵	۶۶/۳۷	۱۳/۷۷	۰/۸۴۴۸	۰/۷۷۲۵	۰/۱۴۸۴
	۶۴	۶۳/۴۹	۵۲/۵۹	۲۰/۱۳	۶۸/۲۸	۱۵/۶۹	۰/۸۴۷۲	۰/۴۰۲۳	۰/۱۰۲۵
	۱۲۸	۶۳/۴۹	۵۲/۵۹	۱۹/۰۴	۶۹/۹۹	۱۷/۴۰	۰/۸۴۳۸	۰/۲۱۳۹	۰/۰۸۷۹
	۲۵۶	۶۳/۴۹	۵۲/۵۹	۱۸/۹۵	۷۰/۱۴	۱۷/۵۴	۰/۸۱۷۴	۰/۱۲۵	۰/۰۸۴
4.1.04	۸	۶۴/۷۸	۳۹/۵۶	۳۶/۱۹	۴۴/۱۲	۴/۵۶	۰/۹۷۱۹	۵/۷۷۲۵	۰/۶۲۷
	۱۶	۶۴/۷۸	۳۹/۵۶	۳۰	۵۳/۶۹	۱۴/۱۲	۰/۹۶۷۵	۲/۴۱۳۱	۰/۲۶۰۷
	۳۲	۶۴/۷۸	۳۹/۵۶	۲۷/۵۷	۵۷/۴۳	۱۷/۸۷	۰/۹۶۲۷	۱/۰۴۹۸	۰/۱۴۶۵
	۶۴	۶۴/۷۸	۳۹/۵۶	۲۶/۴۴	۵۹/۱۷	۱۹/۶۱	۰/۹۶۰۲	۰/۴۳۰۷	۰/۱۰۲۵

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
4.1.05	128	64/78	39/56	24/29	62/49	22/93	0.9474	0.1992	0.0879
	256	64/78	39/56	24/21	62/62	23/06	0.9478	0.1211	0.0840
	8	64/53	43/91	35/06	45/66	1/74	0.9417	5/6875	0.6152
	16	64/53	43/91	30/88	52/14	8/22	0.9327	2/3369	0.2705
	32	64/53	43/91	29/08	54/92	11/01	0.9102	0.9785	0.1553
	64	64/53	43/91	24/95	61/33	17/41	0.8932	0.4316	0.1016
	128	64/53	43/91	23/05	64/27	20/35	0.8716	0.1992	0.0879
4.1.06	256	64/53	43/91	21/88	66/08	22/17	0.8630	0.1211	0.0840
	8	63/50	28/95	47/65	24/96	-3/99	0.9696	9/2178	0.7109
	16	63/50	28/95	41/11	35/25	6/29	0.9666	3/6406	0.2939
	32	63/50	28/95	33/90	46/61	17/65	0.9435	1/3262	0.1514
	64	63/50	28/95	32/91	48/17	19/21	0.9367	0.4316	0.1025
	128	63/50	28/95	29/80	53/07	24/11	0.9359	0.1904	0.0879
	256	63/50	28/95	29/87	52/96	24	0.9325	0.1240	0.0840
4.1.07	8	53/99	52/86	25/60	52/58	-0/27	0.8238	3/8262	0.4951
	16	53/99	52/86	22/53	58/26	5/40	0.7855	1/4883	0.2236
	32	53/99	52/86	21/37	60/40	7/54	0.7154	0.6055	0.1348
	64	53/99	52/86	16/84	68/79	15/92	0.6757	0.3008	0.1016
	128	53/99	52/86	15/78	70/76	17/90	0.5923	0.1777	0.0879
	256	53/99	52/86	15/71	70/89	18/02	0.6075	0.1162	0.0840
	8	55/72	46/12	30/59	45/09	-1/02	0.8788	5/0928	0.5967
4.1.08	16	55/72	46/12	26/34	52/71	6/59	0.8404	2/084	0.251
	32	55/72	46/12	21/70	61/04	14/92	0.7820	0.8018	0.1455
	64	55/72	46/12	20/86	62/55	16/43	0.7594	0.3604	0.1016
	128	55/72	46/12	18/39	66/98	20/86	0.7024	0.1748	0.0879
	256	55/72	46/12	18/33	67/09	20/97	0.7037	0.1172	0.0840
	8	255/99	46/45	120/27	53/01	6/55	0.9596	17/1309	1/9688

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
4.2.01	16	255/99	46/45	104/61	59/13	12/67	0/9519	6/5771	0/6445
	32	255/99	46/45	94/52	63/07	16/61	0/9431	2/4219	0/2627
	64	255/99	46/45	92/47	63/87	27/42	0/9397	0/9629	0/1465
	128	255/99	46/45	85/58	66/56	20/10	0/9404	0/4287	0/1025
	256	255/99	46/45	90/60	64/27	18/14	0/9318	0/1963	0/0879
4.2.02	8	256/33	38/74	191/57	25/26	-13/47	0/9637	200527	20498
	16	256/33	38/74	168/65	34/20	-4/53	0/9587	7/2803	0/6953
	32	256/33	38/74	122/77	52/10	13/35	0/9535	2/6816	0/2705
	64	256/33	38/74	125/56	51/01	12/27	0/9480	1/0068	0/1455
	128	256/33	38/74	128/18	49/99	11/25	0/9408	0/3730	0/1025
	256	256/33	38/74	96/48	62/35	23/61	0/9295	0/1768	0/0879
4.2.03	8	258/56	19/90	196/96	23/82	3/92	0/9268	420332	2/1719
	16	258/56	19/90	166/01	35/79	15/89	0/9959	13/8516	0/6992
	32	258/56	19/90	155/20	39/97	20/07	0/9951	4/1787	0/2598
	64	258/56	19/90	150/72	41/70	21/80	0/9939	1/3057	0/1436
	128	258/56	19/90	145/73	43/63	23/73	0/9927	0/4336	0/1016
	256	258/56	19/90	144/36	44/16	24/26	0/9922	0/1807	0/0879
4.2.04	8	258/38	36/08	159/18	38/39	2/30	0/9726	240693	2/2002
	16	258/38	36/08	126/66	50/97	14/89	0/9670	9/2930	0/75
	32	258/38	36/08	120/27	53/44	17/36	0/9625	3/3564	0/2881
	64	258/38	36/08	116/34	54/97	18/88	0/9592	1/2578	0/1514
	128	258/38	36/08	104/38	59/60	23/51	0/9555	0/4258	0/1025
	256	258/38	36/08	111/33	56/91	20/82	0/9472	0/1855	0/0879
	8	255/51	40/93	170/05	33/44	-7/48	0/9441	23/8320	2/1611
	16	255/51	40/93	136/72	46/49	5/55	0/9347	8/8965	0/7158
	32	255/51	40/93	127/05	50/27	9/34	0/9257	3/0732	0/2783
	64	255/51	40/93	121/88	52/30	11/36	0/919	1/0234	0/1484

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
4.2.05	128	255/51	40/93	112/33	56/03	15/09	0/9044	0/3945	0/1025
	256	255/51	40/93	90/65	66/52	23/58	0/898	0/1885	0/0879
4.2.06	8	258/33	29/30	195/53	24/30	-4/99	0/9827	32/5146	2/3584
	16	258/33	29/30	170/05	34/17	4/86	0/9790	12/3721	0/7744
	32	258/33	29/30	137/48	46/78	17/47	0/9750	4/3555	0/2930
	64	258/33	29/30	137/03	46/95	17/65	0/9686	1/412	0/1504
	128	258/33	29/30	136/10	47/31	18/01	0/9676	0/4697	0/1016
	256	258/33	29/30	120/54	52/33	24/03	0/9654	0/1953	0/0879
4.2.07	8	258/50	35/71	159/75	38/20	2/48	0/9805	25/0654	2/2324
	16	258/50	35/71	129/89	49/75	14/03	0/9764	10/001	0/7793
	32	258/50	35/71	121/18	53/12	17/40	0/9721	3/7881	0/2969
	64	258/50	35/71	116/91	54/77	19/05	0/9650	1/3809	0/1504
	128	258/50	35/71	110/02	55/50	19/78	0/9628	0/4727	0/1025
	256	258/50	35/71	106/26	58/8929	23/17	0/9582	0/1953	0/0879
5.1.09	8	64/83	30/41	40/88	36/93	6/51	0/9931	7/1992	0/6416
	16	64/83	30/41	35/89	44/63	14/21	0/9918	2/7744	0/2676
	32	64/83	30/41	33/89	47/71	17/29	0/9907	1/0811	0/1494
	64	64/83	30/41	32/95	49/17	18/75	0/9896	0/4404	0/1025
	128	64/83	30/41	31/28	51/74	21/32	0/9864	0/1973	0/879
	256	64/83	30/41	30/26	52/46	23/04	0/9860	0/1191	0/0840
5.1.10	8	64/7930	22/33	51/82	20	-2/33	0/9983	11/5790	0/7119
	16	64/7930	22/33	43/14	33/41	11/07	0/9977	4/0107	0/2637
	32	64/7930	22/33	39/46	39/09	16/75	0/9969	1/2227	0/1426
	64	64/7930	22/33	35/15	45/74	23/40	0/9961	0/4023	0/1016
	128	64/7930	22/33	34/84	46/21	23/88	0/9963	0/1885	0/0879
	256	64/7930	22/33	34/92	46/10	23/76	0/9967	0/1201	0/0840

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
5.1.11	8	63/81	49/62	33/43	47/61	-2/0.1	0/90.72	3/90.82	0/5586
	16	63/81	49/62	27/87	56/31	6/69	0/8866	1/6963	0/2510
	32	63/81	49/62	26/45	58/53	8/91	0/8797	0/7959	0/1514
	64	63/81	49/62	20/87	67/28	17/65	0/8736	0/3926	0/1016
	128	63/81	49/62	19/25	69/82	20/19	0/8479	0/1924	0/0879
	256	63/81	49/62	18/50	70/99	21/36	0/8129	0/1240	0/0840
5.1.12	8	63/54	41/19	38/83	38/88	-2/30	0/90.33	5/6133	0/5977
	16	63/54	41/19	34/53	47/23	6/04	0/8929	2/21	0/2529
	32	63/54	41/19	30/81	51/5	10/30	0/8730	0/8418	0/1494
	64	63/54	41/19	30/23	52/42	11/23	0/8481	0/3828	0/1025
	128	63/54	41/19	27/83	56/19	15	0/84	0/1904	0/0879
	256	5410.63	41/19	22/64	64/35	23/16	0/80.88	0/1211	0/0840
5.1.13	8	13/30	9/0.1	13/47	-1/12	7/89	0/8604	3/4990	0/4922
	16	13/30	9/0.1	13/23	0/63	8/38	0/8602	1/9287	0/2510
	32	13/30	9/0.1	12/45	8/54	0/47	0/8881	1/0645	0/1504
	64	13/30	9/0.1	12/21	12/42	3/40	0/9463	0/4941	0/1025
	128	13/30	9/0.1	12/10	15/83	6/81	0/9775	0/2041	0/0879
	256	13/30	9/0.1	12/22	12/44	3/42	0/9959	0/1260	0/0840
5.1.14	8	64/70	29/26	43/26	33/13	3/87	0/9957	9/4297	0/6999
	16	64/70	29/26	36/32	43/86	14/60	0/9944	4/5127	0/2822
	32	64/70	29/26	33/43	48/32	19/06	0/9931	1/3164	0/1523
	64	64/70	29/26	31/73	9599	21/69	0/9908	0/4668	0/1016
	128	64/70	29/26	30/43	52/95	23/69	0/9898	0/2041	0/0879
	256	64/70	29/26	30/53	52/81	23/55	0/9906	0/1201	0/0840
	8	257/86	35/97	153/08	40/63	4/65	0/9829	26/2724	2/1494
	16	257/86	35/97	124/93	51/54	15/56	0/9814	9/6318	0/709

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
5.2.08	۳۲	۲۵۷/۸۶	۳۵/۹۷	۱۱۶/۷۳	۵۴/۷۳	۱۸/۷۵	۰/۹۷۸۹	۳/۳۹۴۵	۰/۲۷۵۴
	۶۴	۲۵۷/۸۶	۳۵/۹۷	۱۱۳/۵۰	۵۵/۹۸	۲۰	۰/۹۷۸۵	۱/۲۳۶۳	۰/۱۴۷۵
	۱۲۸	۲۵۷/۸۶	۳۵/۹۷	۱۰۹/۷۲	۵۷/۴۴	۲۱/۴۷	۰/۹۷۷۱	۰/۴۶	۰/۱۰۲۵
	۲۵۶	۲۵۷/۸۶	۳۵/۹۷	۱۱۰/۶۰	۵۷/۱۰	۲۱/۱۲	۰/۹۷۶۵	۰/۲۰۳۱	۰/۰۸۷۹
5.2.09	۸	۲۵۷/۱۳	۲۸/۱۹	۱۶۰/۱۴	۳۷/۷۱	۹/۵۱	۰/۹۸۲۳	۳۵/۸۳۰۱	۰/۵۶۵۴
	۱۶	۲۵۷/۱۳	۲۸/۱۹	۱۳۶/۳۹	۴۶/۹۵	۱۸/۷۵	۰/۹۸۰۳	۱۳/۸۸۳۸	۰/۷۸۳۲
	۳۲	۲۵۷/۱۳	۲۸/۱۹	۱۲۶/۶۴	۵۰/۷۴	۲۲/۵۴	۰/۹۷۹۰	۴/۶۴۶۵	۰/۲۷۶۴
	۶۴	۲۵۷/۱۳	۲۸/۱۹	۱۲۲/۳۰	۵۲/۴۳	۲۴/۲۳	۰/۹۷۸۴	۰/۴۸۰۵	۰/۱۰۱۶
	۱۲۸	۲۵۷/۱۳	۲۸/۱۹	۱۲۲	۵۲/۵۵	۲۴/۳۵	۰/۹۷۹۴	۰/۱۹۹۲	۰/۰۸۷۹
	۲۵۶	۲۵۷/۱۳	۲۸/۱۹	۱۴۳/۷۲	۴۴/۱۰	۱۵/۹۰	۰/۹۸۲۲	۱/۴۹۶۱	۰/۱۴۹۴
5.2.10	۸	۲۴۷/۸۳	۲۵/۵۹	۱۷۰/۶۵	۳۱/۱۴	۵/۵۴	۰/۹۹۹۹	۲۱/۳۹۰۶	۲/۲۶۵۶
	۱۶	۲۴۷/۸۳	۲۵/۵۹	۱۵۲/۰۴	۳۸/۶۴	۱۳/۰۵	۰/۹۹۹	۶/۲۲۲۷	۰/۷۱۵۸
	۳۲	۲۴۷/۸۳	۲۵/۵۹	۱۴۸/۰۸	۴۰/۲۴	۱۴/۶۵	۰/۹۹۹	۱/۸۵۲۵	۰/۲۶۶۶
	۶۴	۲۴۷/۸۳	۲۵/۵۹	۱۴۳/۵۷	۴۲/۰۶	۱۶/۴۷	۰/۹۹۹۹	۰/۶۱۴۳	۰/۱۴۱۶
	۱۲۸	۲۴۷/۸۳	۲۵/۵۹	۱۳۹/۴۱	۴۳/۷۴	۲۰/۱۵	۰/۹۹۹۹	۰/۲۷۵۴	۰/۱۰۱۶
	۲۵۶	۲۴۷/۸۳	۲۵/۵۹	۱۳۳/۴۷	۴۶/۱۴	۲۰/۵۴	۰/۹۹۹۹	۰/۱۴۴۵	۰/۰۸۷۹
5.3.01	۸	۱۰۲۴	۳۲/۵۰	۶۴/۹۹	۳۷/۵۵	۵/۰۴	۰/۹۷۹۱	۱۱۱/۵۸۳۰	۸/۶۱۶۲
	۱۶	۱۰۲۴	۳۲/۵۰	۵۵۰/۰۷	۴۶/۴۱	۱۳/۹۰	۰/۹۷۶۶	۴۱/۳۸۳۸	۲/۶۲۱۱
	۳۲	۱۰۲۴	۳۲/۵۰	۴۹۶/۲۸	۵۱/۶۵	۱۹/۱۴	۰/۹۷۴۷	۱۴/۳۱۱۵	۰/۸۲۱۳
	۶۴	۱۰۲۴	۳۲/۵۰	۴۷۹/۸۱	۵۳/۲۵	۲۰/۱۵	۰/۹۷۴۱	۴/۸۱۹۳	۰/۲۹۳۹
	۱۲۸	۱۰۲۴	۳۲/۵۰	۴۶۷/۳۵	۵۴/۴۷	۲۱/۹۶	۰/۹۷۲۲	۱/۴۹۹۰	۰/۱۵۱۴
	۲۵۶	۱۰۲۴	۳۲/۵۰	۴۵۸/۳۴	۵۵/۳۴	۲۲/۸۴	۰/۹۷۳۴	۰/۴۷۷۵	۰/۱۰۱۶
5.3.02	۸	۱۰۳۳	۲۷/۵۷	۶۸۳/۶۹	۳۴/۸۴	۶/۲۷	۰/۹۹۲۶	۱۲۹/۱۸۱۶	۸/۷۲۷۵
	۱۶	۱۰۳۳	۲۷/۵۷	۵۹۲/۳۲	۴۲/۶۸	۱۵/۱۱	۰/۹۹۱۱	۴۷/۰۶۹۳	۲/۶۴۹۴
	۳۲	۱۰۳۳	۲۷/۵۷	۵۵۶/۱۲	۴۶/۱۸	۱۸/۶۱	۰/۹۸۹۸	۱۵/۷۹۲۰	۰/۷۹۶۹
	۶۴	۱۰۳۳	۲۷/۵۷	۵۳۸/۸۹	۴۷/۸۵	۲۰/۲۸	۰/۹۸۸۶	۴/۹۱۶۵	۰/۲۹۵۹
	۱۲۸	۱۰۳۳	۲۷/۵۷	۵۳۳	۴۸/۴۲	۲۰/۸۵	۰/۹۸۸۲	۱/۴۷۷۵	۰/۱۵۰۴

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
	۲۵۶	۱۰۳۳	۲۷/۵۷	۵۲۸/۹۲	۴۸/۸۱۹	۲۱/۲۴	۰/۹۸۷۲	۰/۴۸۷۳	۰/۱۰۱۶
7.1.01	۸	۲۵۷/۴۸	۳۹/۳۰	۱۳۹/۰۶	۴۵/۹۹	۶/۶۸	۰/۹۹۰۷	۲۵/۳۲۶۲	۲/۱۵۰۴
	۱۶	۲۵۷/۴۸	۳۹/۳۰	۱۱۸/۸۸	۵۳/۸۳	۱۴/۵۲	۰/۹۸۸۷	۸/۸۸۴۸	۰/۶۷۷۷
	۳۲	۲۵۷/۴۸	۳۹/۳۰	۱۰۹/۲۲	۵۷/۵۷	۱۸/۲۷	۰/۹۸۵۹	۲/۸۷۲۱	۰/۲۶۱۷
	۶۴	۲۵۷/۴۸	۳۹/۳۰	۱۰۵/۴۵	۵۹/۰۴	۱۹/۷۳	۰/۹۸۴۰	۰/۹۵۶۱	۰/۱۴۰۶
	۱۲۸	۲۵۷/۴۸	۳۹/۳۰	۱۰۲/۹۷	۶۰	۲۰/۷۰	۰/۹۸۲۵	۰/۳۵۵۵	۰/۱۰۱۶
	۲۵۶	۲۵۷/۴۸	۳۹/۳۰	۹۸/۲۲	۶۱/۸۵	۲۲/۵۴	۰/۹۷۹۸	۰/۱۶۸۹	۰/۰۸۷۹
7.1.02	۸	۲۳۳/۸۳	۵۳/۹۶	۱۳۳/۹۵	۴۰/۱۵	-۱۳/۸۱	۰/۹۹۸۷	۱۲/۰۹۷۷	۰/۸۵۴۵
	۱۶	۲۳۳/۸۳	۵۳/۹۶	۱۲۰/۶۵	۴۶/۰۹	-۷/۸۶	۰/۹۹۷۷	۴/۷۸۵۲	۰/۶۴۳۶
	۳۲	۲۳۳/۸۳	۵۳/۹۶	۸۲/۳۷	۶۳/۱۹	۹/۲۳	۰/۹۹۵۳	۱/۹۵۵۱	۰/۲۶۹۵
	۶۴	۲۳۳/۸۳	۵۳/۹۶	۸۰/۳۶	۶۴/۰۹	۱۰/۱۳	۰/۹۹۰۸	۰/۸۴۵۷	۰/۱۵۱۴
	۱۲۸	۲۳۳/۸۳	۵۳/۹۶	۷۹/۳۵	۶۴/۵۴	۱۰/۵۸	۰/۹۷۹۸	۰/۳۵۵۵	۰/۱۰۱۶
	۲۵۶	۲۳۳/۸۳	۵۳/۹۶	۸۰/۵۷	۶۴	۱۰/۰۳	۰/۹۹۸۹	۰/۱۸۸۵	۰/۰۸۷۹
7.1.03	۸	۲۵۸/۳۱	۳۷/۳۱	۱۴۴/۸۵	۴۳/۹۲	۶/۶۰	۰/۹۹۳۰	۲۳/۸۳۴	۲/۰۶۷۴
	۱۶	۲۵۸/۳۱	۳۷/۳۱	۱۲۶/۶۸	۵۰/۹۵	۱۳/۶۳	۰/۹۹۱۸	۸/۲۸۰۳	۰/۶۶۰۲
	۳۲	۲۵۸/۳۱	۳۷/۳۱	۱۱۹/۶۹	۵۳/۶۶	۱۶/۳۴	۰/۹۹۰۵	۲/۸۰۸۶	۰/۲۵۲۹
	۶۴	۲۵۸/۳۱	۳۷/۳۱	۱۱۶/۲۸	۵۴/۹۸	۱۷/۶۶	۰/۹۸۹۶	۰/۹۲۸۷	۰/۱۴۳۶
	۱۲۸	۲۵۸/۳۱	۳۷/۳۱	۱۱۳/۰۴	۵۶/۲۳	۱۸/۹۱	۰/۹۸۸۳	۰/۳۷۰۱	۰/۱۰۱۶
	۲۵۶	۲۵۸/۳۱	۳۷/۳۱	۱۱۰/۱۴	۵۷/۳۵	۲۰/۰۴	۰/۹۸۶۱	۰/۱۸۰۷	۰/۰۸۷۹
7.1.04	۸	۱۵۸/۰۸	۳۹/۸۰	۱۳۸/۳۰	۴۶/۴۰	۶/۶۰	۰/۹۹۱۵	۰/۰۰۷۸	۲/۰۸۲۰
	۱۶	۱۵۸/۰۸	۳۹/۸۰	۱۲۰/۲۵	۵۳/۴۰	۱۳/۵۹	۰/۹۹	۸/۴۸۸۳	۰/۶۶۳۱
	۳۲	۱۵۸/۰۸	۳۹/۸۰	۱۱۲/۲۹	۵۶/۴۸	۱۶/۶۸	۰/۹۸۶۳	۲/۸۱۷۴	۰/۲۵
	۶۴	۱۵۸/۰۸	۳۹/۸۰	۱۰۷/۷۰	۵۸/۲۶	۱۸/۴۶	۰/۹۸۴۷	۰/۹۴۸۲	۰/۱۴۲۶
	۱۲۸	۱۵۸/۰۸	۳۹/۸۰	۱۰۳/۰۵	۶۰/۰۷	۲۰/۲۶	۰/۹۸۲۳	۰/۳۵۹۴	۰/۱۰۱۶
	۲۵۶	۱۵۸/۰۸	۳۹/۸۰	۹۹/۱۷	۶۱/۷۶	۲۱/۷۶	۰/۹۸۱۵	۰/۱۸۲۶	۰/۰۸۷۹
	۸	۲۵۸/۳۲	۳۰/۲۳	۱۶۹/۱۴	۳۴/۵۲	۴/۲۹	۰/۹۹۴۱	۳۲/۹۴۹۲	۲/۱۷۵۸
	۱۶	۲۵۸/۳۲	۳۰/۲۳	۱۴۱/۸۰	۴۵/۱۰	۱۴/۸۷	۰/۹۹۲۶	۱۱/۵۴۵۹	۰/۷۰۵۱

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
7.1.05	۳۲	۲۵۸/۳۲	۳۰/۲۳	۱۳۰/۸۵	۴۹/۳۴	۱۹/۱۱	۰/۹۹۱۱	۳/۶۳۴۸	۰/۲۶۹۵
	۶۴	۲۵۸/۳۲	۳۰/۲۳	۱۲۵/۹۶	۵۱/۲۳	۲۱	۰/۹۸۷۲	۱/۱۳۳۸	۰/۱۴۵۵
	۱۲۸	۲۵۸/۳۲	۳۰/۲۳	۱۲۰/۴۸	۵۳/۳۵	۲۳/۱۲	۰/۹۸۴۳	۰/۳۷۴۰	۰/۱۰۱۶
	۲۵۶	۲۵۸/۳۲	۳۰/۲۳	۱۲۲/۵۷	۵۲/۵۴	۲۲/۳۱	۰/۹۸۴۱	۰/۱۷۹۷	۰/۰۸۷۹
7.1.06	۸	۲۵۸/۵۲	۳۰/۳۴	۱۶۶/۷۲	۳۵/۵۱	۵/۱۶	۰/۹۹۴۶	۳۳/۱۷۶۸	۲/۱۴۱۶
	۱۶	۲۵۸/۵۲	۳۰/۳۴	۱۴۰/۸۷	۴۵/۵۰	۱۵/۱۶	۰/۹۹۳۱	۱۱/۵۱۴۶	۰/۶۸۲۸
	۳۲	۲۵۸/۵۲	۳۰/۳۴	۱۳۰/۱۵	۴۹/۶۵	۱۹/۳۱	۰/۹۹۱۹	۳/۵۷۲۳	۰/۲۵۷۸
	۶۴	۲۵۸/۵۲	۳۰/۳۴	۱۲۵/۰۴	۵۱/۶۳	۲۳/۲۹	۰/۹۹۱۴	۱/۱۰۰۶	۰/۱۴۰۶
	۱۲۸	۲۵۸/۵۲	۳۰/۳۴	۱۲۰/۰۶	۵۳/۵۵	۲۳/۲۱	۰/۹۸۸۷	۰/۳۵۳۵	۰/۱۰۱۶
	۲۵۶	۲۵۸/۵۲	۳۰/۳۴	۱۲۰/۲۶	۵۳/۴۷	۲۳/۱۳	۰/۹۸۹۲	۰/۱۷۱۹	۰/۰۸۷۹
7.1.07	۸	۲۵۸/۵۱	۳۰/۰۸	۱۶۵/۴۴	۳۶	۵/۹۲	۰/۹۹۷۰	۳۰/۴۹۷۱	۲/۰۸۵۰
	۱۶	۲۵۸/۵۱	۳۰/۰۸	۱۴۲/۵۸	۴۴/۸۴	۱۴/۷۶	۰/۹۹۶۰	۱۰/۵۰۶۸	۰/۶۵۲۳
	۳۲	۲۵۸/۵۱	۳۰/۰۸	۱۳۳/۳۰	۴۸/۴۳	۱۸/۳۵	۰/۹۹۵۲	۳/۳۳۳۰	۰/۲۴۸۰
	۶۴	۲۵۸/۵۱	۳۰/۰۸	۱۲۸/۰۱	۵۰/۴۸	۲۰/۴۰	۰/۹۹۳۸	۱/۰۲۵۴	۰/۱۴۱۶
	۱۲۸	۲۵۸/۵۱	۳۰/۰۸	۱۲۵/۱۶	۵۱/۵۸	۲۱/۵۰	۰/۹۹۲۸	۰/۳۷۶۰	۰/۱۰۲۵
	۲۵۶	۲۵۸/۵۱	۳۰/۰۸	۱۲۱/۶۱	۵۲/۹۲	۲۲/۸۷	۰/۹۹۱۱	۰/۱۷۷۷	۰/۰۸۷۹
7.1.08	۸	۲۵۷/۵۰	۴۲/۳۲	۱۲۸/۰۱	۵۰/۲۸	۷/۹۶	۰/۹۸۵۱	۱۹/۴۵۲۱	۱/۹۷۱۷
	۱۶	۲۵۷/۵۰	۴۲/۳۲	۱۱۳/۴۹	۵۵/۹۲	۱۳/۶۰	۰/۹۸۳۴	۷/۰۳۰۳	۱/۶۴۴۵
	۳۲	۲۵۷/۵۰	۴۲/۳۲	۱۰۸/۰۱	۵۸/۷۳	۱۵/۷۳	۰/۹۸۲۴	۲/۴۳۹۵	۰/۲۵۴۹
	۶۴	۲۵۷/۵۰	۴۲/۳۲	۱۰۳/۴۹	۵۹/۸۰	۱۷/۴۸	۰/۹۸۲۱	۰/۸۷۴۰	۰/۱۴۳۶
	۱۲۸	۲۵۷/۵۰	۴۲/۳۲	۱۰۵/۹۰	۵۸/۸۷	۱۶/۵۵	۰/۹۷۸۴	۰/۳۵۱۶	۰/۱۰۱۶
	۲۵۶	۲۵۷/۵۰	۴۲/۳۲	۱۰۲/۵۵	۶۰/۱۷	۱۷/۸۵	۰/۹۷۹۷	۰/۱۸۴۶	۰/۰۸۷۹
7.1.09	۸	۲۵۸/۳۴	۳۵/۵۶	۱۵۴/۱۲	۴۰/۳۴	۴/۷۷	۰/۹۹۶۸	۲۸/۸۵۲۵	۲/۰۹۰۸
	۱۶	۲۵۸/۳۴	۳۵/۵۶	۳۲/۵۱	۴۸/۷۰	۱۳/۱۴	۰/۹۹۶۰	۹/۹۱۳۱	۰/۶۶۱۱
	۳۲	۲۵۸/۳۴	۳۵/۵۶	۱۲۳/۸۷	۵۲/۰۵	۱۶/۴۸	۰/۹۹۵۱	۳/۲۲۴۶	۰/۲۵
	۶۴	۲۵۸/۳۴	۳۵/۵۶	۱۱۹/۱۶	۵۳/۸۷	۱۸/۳۰	۰/۹۹۴۰	۱/۰۰۱	۰/۱۳۹۶
	۱۲۸	۲۵۸/۳۴	۳۵/۵۶	۱۱۶/۲۵	۵۵	۱۹/۴۳	۰/۹۹۳۱	۰/۳۷۶۰	۰/۱۰۱۶

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
	۲۵۶	۲۵۸/۳۴	۳۵/۵۶	۱۱۰/۵۷	۵۷/۱۹	۲۱/۶۳	۰/۹۹۲۳	۰/۱۸۰۷	۰/۰.۸۷۹
7.1.10	۸	۲۵۸/۴۹	۳۹/۰.۷	۱۴۱/۳۲	۴۵/۳۲	۶/۲۵	۰/۹۹۴۶	۲۴/۶۰.۴۵	۲/۰.۸۴۰
	۱۶	۲۵۸/۴۹	۳۹/۰.۷	۱۲۲/۷۶	۵۲/۵۰	۱۳/۴۳	۰/۹۹۳۲	۸/۶۴۱۶	۰/۶۴۸۴
	۳۲	۲۵۸/۴۹	۳۹/۰.۷	۱۱۵/۱۵	۵۵/۴۵	۱۶/۲۷	۰/۹۹۲۰	۲/۹۴۶۳	۰/۲۵۸۸
	۶۴	۲۵۸/۴۹	۳۹/۰.۷	۱۱۰/۴۸	۵۷/۲۵	۱۸/۱۸	۰/۹۹۰.۴	۰/۹۵۱۲	۰/۱۳۸۷
	۱۲۸	۲۵۸/۴۹	۳۹/۰.۷	۱۰۰/۹۰	۶۰/۹۶	۲۱/۸۸	۰/۹۸۹۳	۰/۳۵۷۴	۰/۱۰۰.۱۶
	۲۵۶	۲۵۸/۴۹	۳۹/۰.۷	۱۰۶/۴۱	۵۸/۸۳	۱۹/۷۵	۰/۹۸۷۳	۰/۱۶۵	۰/۰.۸۷۹
7.2.01	۸	۱۰.۳۳	۳۹/۶۳	۵۱۱/۰.۵	۵۰/۵۴	۱۰/۹۰	۰/۹۷۹۰	۶۹/۳۲۷۱	۶/۶۷۲۹
	۱۶	۱۰.۳۳	۳۹/۶۳	۴۵۵/۲۴	۵۵/۹۴	۱۶/۳۰	۰/۹۷۵۵	۲۲/۳۱۵۴	۱/۹۵۹۰
	۳۲	۱۰.۳۳	۳۹/۶۳	۴۳۵/۵۳	۵۷/۸۴	۱۸/۲۰	۰/۹۷۳۰	۷/۱۸۸۵	۰/۶۴۸۴
	۶۴	۱۰.۳۳	۳۹/۶۳	۴۲۸/۰.۴	۵۸/۵۷	۱۸/۹۳	۰/۹۷۱۶	۲/۴۲۴۸	۰/۲۶۵۶
	۱۲۸	۱۰.۳۳	۳۹/۶۳	۴۲۹/۰.۶	۵۸/۴۷	۱۸/۸۳	۰/۹۷۲۰	۰/۰.۸۷۶	۰/۱۵۰.۴
	۲۵۶	۱۰.۳۳	۳۹/۶۳	۴۲۶/۰.۴	۵۸/۷۶	۱۹/۱۲	۰/۹۷۱۲	۰/۳۳۶۹	۰/۱۰.۲۵
boat	۸	۲۵۸/۴۵	۲۹/۴۹	۱۶۸/۳۰	۳۴/۸۷	۵/۳۸	۰/۹۸۷۱	۳۰/۳۴۱۸	۲/۲۹۱۰
	۱۶	۲۵۸/۴۵	۲۹/۴۹	۱۴۵/۱۴	۴۳/۸۴	۱۴/۳۴	۰/۹۸۴۴	۱۱/۵۰.۳۹	۰/۷۶۱۷
	۳۲	۲۵۸/۴۵	۲۹/۴۹	۱۳۴/۵۸	۴۷/۹۲	۱۸/۴۳	۰/۹۸۱۲	۴/۰۰.۷۸	۰/۲۹۲۰
	۶۴	۲۵۸/۴۵	۲۹/۴۹	۱۳۰/۶۳	۴۹/۴۵	۱۹/۹۲	۰/۹۷۸۲	۱/۳۶۱۳	۰/۱۵۲۳
	۱۲۸	۲۵۸/۴۵	۲۹/۴۹	۱۲۷/۲۲	۵۰/۷۷	۲۱/۲۸	۰/۹۷۳۶	۰/۴۸۰.۵	۰/۱۰.۲۵
	۲۵۶	۲۵۸/۴۵	۲۹/۴۹	۱۲۷/۸۶	۵۰/۵۲	۲۱/۰.۳	۰/۹۷۴۶	۰/۲۰.۴۱	۰/۰.۸۷۹
elaine	۸	۲۵۷/۳۷	۳۲/۴۵	۱۷۷/۷۹	۳۰/۹۱	-۱/۵۳	۰/۹۸۹۱	۲۶/۰۰.۷۸	۲/۱۷۹۷
	۱۶	۲۵۷/۳۷	۳۲/۴۵	۱۵۹/۱۴	۳۸/۱۶	۵/۷۰	۰/۹۸۵۸	۹/۷۴۵۱	۰/۷۳۴۴
	۳۲	۲۵۷/۳۷	۳۲/۴۵۸	۱۳۰/۸۷	۴۹/۱۴	۱۶/۶۹	۰/۹۸۲۴	۳/۴۸۷۳	۰/۲۸۹۱
	۶۴	۲۵۷/۳۷	۳۲/۴۵	۱۲۷/۰.۸	۵۰/۶۲	۱۸/۱۶	۰/۹۸۰.۱	۱/۲۳۳۴	۰/۱۵۳۳
	۱۲۸	۲۵۷/۳۷	۳۲/۴۵	۱۲۸/۲۲	۵۰/۱۷	۱۷/۷۲	۰/۹۷۸۶	۰/۴۲۶۸	۰/۱۰.۱۶
	۲۵۶	۲۵۷/۳۷	۳۲/۴۵	۱۱۴/۰.۱	۵۵/۶۹	۲۳/۲۴	۰/۹۷۵۵	۰/۱۹۷۳	۰/۰.۸۷۹
	۸	۶/۴۴	۲۰/۵۵	۶/۵۲	-۱/۲۴	-۲۱/۷۹	۰/۹۵۵۳	۰/۹۰.۹۲	۰/۱۵۰.۴
	۱۶	۶/۴۴	۲۰/۵۵	۵/۹۱	۸/۱۹	-۱۲/۳۵	۰/۹۵۰.۴	۰/۴۸۹۳	۰/۱۲۴۰

All databas	Block dimension	Original size (Kb)	CR-loosless JPEG (%)	Size Compress + overhead (Kb)	CR purposed (%)	Improve (%)	SSIM	Overhead1 (Kb)	Overhead2 (Kb)
Gray21	۳۲	۶/۴۴	۲۰/۵۵	۴/۸۹	۲۴/۱۰	۴/۴۶	۰/۹۵۰۴	۰/۳۱۰۵	۰/۱۱۷۲
	۶۴	۶/۴۴	۲۰/۵۵	۴/۱۱	۳۶/۱۸	۱۵/۶۳	۰/۹۵۱۳	۰/۲۰۹۰	۰/۱۰۹۴
	۱۲۸	۶/۴۴	۲۰/۵۵	۳/۱۳۶	۵۱/۳۲	۳۰/۷۷	۰/۹۵۶۴	۰/۱۳۶۷	۰/۱۰۱۶
	۲۵۶	۶/۴۴	۲۰/۵۵	۳/۰۱۲	۵۳/۲۵	۳۲/۷	۰/۹۵۶۴	۰/۱۱۱۳	۰/۰۸۷۹
house	۸	۱۸۸/۵۰	۷۸/۴۱	۱۷۸/۶۶	۲۸/۳۴	-۴/۵۷	۰/۹۴۸	۲۸/۶۵۰۴	۲/۱۱۵۲
	۱۶	۱۸۸/۵۰	۷۸/۴۱	۱۵۷/۵۲	۳۶/۸۲	۳/۸۲	۰/۹۴۳۳	۱۰/۴۱۸	۰/۷۱۴۸
	۳۲	۱۸۸/۵۰	۷۸/۴۱	۱۴۶/۶۹	۴۱/۱۶	۸/۲۴	۰/۹۲۳۴	۳/۴۵۲۱	۰/۲۷۴۴
	۶۴	۱۸۸/۵۰	۷۸/۴۱	۱۱۸/۴۴	۵۲/۴۹	۱۹/۵۷	۰/۹۱۸۹	۱/۲۲۷۵	۰/۱۴۸۴
	۱۲۸	۱۸۸/۵۰	۷۸/۴۱	۱۱۸/۲۸	۵۲/۵۶	۱۹/۶۴	۰/۹۰۹۴	۰/۴۴۰۴	۰/۱۰۱۶
	۲۵۶	۱۸۸/۵۰	۷۸/۴۱	۱۰۷/۴۰	۵۶/۹۲	۲۴	۰/۸۹۷۷	۰/۱۸۷۵	۰/۰۸۷۹
number	۸	۵۵/۸۱	۲۲/۸۵	۱۷۰/۲۰	۲۶/۴۹	۳/۶۴	۰/۹۹۰۷	۲۶/۱۴۴۵	۲/۴۱۹۹
	۱۶	۵۵/۸۱	۲۲/۸۵	۱۴۴/۵۰	۳۷/۵۹	۱۴/۷۴	۰/۹۸۷۰	۱۱/۵۴۸۸	۰/۷۵۱
	۳۲	۵۵/۸۱	۲۲/۸۵	۱۳۴/۸۹	۴۱/۷۴	۱۸/۸۹	۰/۹۸۵۲	۳/۹۹۸	۰/۲۹۳۹
	۶۴	۵۵/۸۱	۲۲/۸۵	۱۳۰/۳۹	۴۳/۶۹	۲۰/۸۳	۰/۹۸۳۷	۱/۴۳۳۶	۰/۱۴۹۴
	۱۲۸	۵۵/۸۱	۲۲/۸۵	۱۲۶/۴۳	۴۵/۴	۲۲/۵۴	۰/۹۸۲۷	۰/۴۸۴۴	۰/۱۰۱۶
	۲۵۶	۵۵/۸۱	۲۲/۸۵	۱۲۰/۲۷	۴۸/۰۶	۲۵/۲۱	۰/۹۸۱۰	۰/۱۹۶۳	۰/۰۸۷۹
ruler	۸	۲۳۱/۵۷	۱۳/۴۶	۲۱۱/۸۱	۸/۵۳	-۴/۹۳	۰/۷۶۱۹	۱/۵۳۷۱	۰/۲۲۹۵
	۱۶	۲۳۱/۵۷	۱۳/۴۶	۱۹۸/۳۱	۱۴/۳۶	۱/۹۰	۰/۹۷۲۱	۰/۳۵۹۴	۰/۱۵۰۴
	۳۲	۲۳۱/۵۷	۱۳/۴۶	۱۶۴/۹۰	۲۸/۷۹	۱۵/۳۳	۰/۹۷۶۷	۰/۱۹۱۴	۰/۱۰۲۵
	۶۴	۲۳۱/۵۷	۱۳/۴۶	۱۳۹/۹۶	۳۹/۵۶	۲۶/۱	۰/۹۹۹۹	۰/۹۳۸	۰/۰۸۹۸
	۱۲۸	۲۳۱/۵۷	۱۳/۴۶	۱۳۱/۳۷	۴۳/۲۷	۲۹/۳۱	۰/۹۹۹۹	۰/۰۹۶۷	۰/۰۸۹۸
	۲۵۶	۲۳۱/۵۷	۱۳/۴۶	۱۲۴/۰۵	۲۸/۱۲	۳۲/۹	۰/۹۹۹۹	۰/۰۸۷۹	۰/۰۸۷۹
testpat	۸	۷۷۵/۰۲	۱۴/۹۵	۶۹۴/۹۹	۱۰/۳۲	-۴/۶۲	۰/۹۶۲۳	۱۶/۵۸۶۹	۱/۲۱۵۸
	۱۶	۷۷۵/۰۲	۱۴/۹۵	۶۷۶/۲۸	۱۲/۷۴	-۲/۲۱	۰/۹۶	۶/۸۳۲	۰/۵۵۹۶
	۳۲	۷۷۵/۰۲	۱۴/۹۵	۶۵۵/۱۹	۱۵/۴۶	۰/۵۱۱	۰/۹۵۹۱	۲/۶۰۲۵	۰/۳۳۱۱
	۶۴	۷۷۵/۰۲	۱۴/۹۵	۶۴۴/۸۳	۱۹/۳۷	۴/۴۲	۰/۹۵۵	۱/۰۳۲۲	۰/۱۹۷۳
	۱۲۸	۷۷۵/۰۲	۱۴/۹۵	۵۷۰/۲۶	۲۶/۴۲	۱۱/۴۷	۰/۹۵۷۶	۰/۴۲۰۹	۰/۱۳۰۹
	۲۵۶	۷۷۵/۰۲	۱۴/۹۵	۵۵۷/۰۸	۲۸/۱۲	۱۳/۱۷	۰/۹۵۶۸	۰/۲۴۷۱	۰/۱۰۱۶

جدول پ-۲-۲: نتایج حاصل از پیش‌پردازش سوم (دسته‌بندی سطوح خاکستری و متراکم کردن هیستوگرام بر اساس سیستم بینایی انسان) در بهبود عملکرد JPEG2000 بر روی تصاویر پایگاه داده USD.

All database	Block dimension	CR-JPEG2000 (%)	CR-purposed (%)	Improvement (%)	SSIM
4.1.01	۱۲۸	۴۷/۷۴	۶۶/۹۴	۱۹/۲	۰/۹۶۲۲
4.1.02	۱۲۸	۴۸/۴۹	۶۵/۴	۱۶/۹۱	۰/۹۳۱۵
4.1.03	۱۲۸	۳۲/۲۷	۷۳/۲۲	۴۰/۹۵	۰/۸۴۳۸
4.1.04	۱۲۸	۵۲/۶	۷۰/۷۱	۱۸/۱۱	۰/۹۴۷۴
4.1.05	۲۵۶	۵۱/۳۴	۷۰/۶	۱۹/۲۶	۰/۹۳۲۵
4.1.06	۱۲۸	۳۴/۱۷	۵۷/۳۸	۲۳/۲۱	۰/۹۳۵۹
4.1.07	۱۲۸	۵۹/۸۶	۷۹/۳۸	۱۹/۵۲	۰/۵۹۲۳
4.1.08	۱۲۸	۵۸	۷۵/۸۱	۱۷/۸۱	۰/۷۰۲۴
4.2.01	۲۵۶	۵۳/۱۲	۷۰/۹	۱۷/۷۸	۰/۹۳۱۸
4.2.02	۲۵۶	۴۷/۳۳	۷۰/۰۱	۲۲/۶۸	۰/۹۲۲۵
4.2.03	۲۵۶	۲۴/۹۶	۴۹/۹۹	۲۵/۰۳	۰/۹۲۹۵
4.2.04	۲۵۶	۴۶/۹۷	۶۸/۶۸	۲۱/۷۱	۰/۹۹۲۲
4.2.05	۲۵۶	۲۰/۲۹	۷۰/۹۵	۵۰/۶۶	۰/۹۵۵۵
4.2.06	۲۵۶	۳۶/۵۱	۶۰/۰۴	۲۳/۵۳	۰/۸۹۸
4.2.07	۲۵۶	۴۳/۱۳	۶۶/۶۴	۲۳/۵۱	۰/۹۶۵۴
5.1.09	۲۵۶	۲۷/۵	۶۰/۶۶	۳۳/۱۶	۰/۹۵۸۲
5.1.10	۶۴	۲۷/۴۵	۵۰/۴۸	۲۳/۰۳	۰/۹۸۶
5.1.11	۲۵۶	۵۷/۴۶	۷۵/۶۴	۱۸/۱۸	۰/۹۹۶۱
5.1.12	۶۴	۵۲/۷۸	۷۲/۷۶	۱۹/۹۸	۰/۸۱۲۹
5.1.13	۲۵۶	۲۲/۱۵	۲۷/۸۱	۵/۶۶	۰/۸۰۸۸
5.1.14	۱۲۸	۲۵/۷	۵۸/۲۶	۲۳/۰۹	۰/۹۴۶۳
5.2.08	۱۲۸	۴۴/۹۸	۶۴/۴۵	۱۹/۴۷	۰/۹۹۰۶
5.2.09	۲۵۶	۳۵/۸۳	۵۸/۶۶	۲۳/۸۳	۰/۹۷۷۱
5.2.10	۲۵۶	۲۶/۱۵	۵۲/۲۹	۲۶/۱۴	۰/۹۷۹۴
5.3.01	۲۵۶	۳۹/۷۴	۶۲/۴۴	۲۲/۷	۰/۹۹۹
5.3.02	۲۵۶	۳۲/۲۳	۵۷/۳۴	۲۵/۱۱	۰/۹۷۳۴
7.1.01	۲۵۶	۴۱/۳۵	۶۴/۵۵	۲۳/۲	۰/۹۸۷۲
7.1.02	۲۵۶	۴۷/۲۸	۶۶/۴۱	۱۹/۱۳	۰/۹۷۹۸
7.1.03	۲۵۶	۳۸/۸۳	۶۱/۴۹	۲۲/۶۶	۰/۹۹۸۹
7.1.04	۲۵۶	۵۳/۱۱	۶۵/۸۳	۱۲/۷۲	۰/۹۸۶۱
7.1.05	۱۲۸	۳۳/۰۲	۵۷/۳۱	۲۴/۲۹	۰/۹۸۱۵
7.1.06	۲۵۶	۳۳/۰۸	۵۷/۶۶	۲۴/۵۸	۰/۹۸۴۳
7.1.07	۱۲۸	۳۳/۸۵	۵۹/۳۷	۲۵/۴۲	۰/۹۸۸۷
7.1.08	۲۵۶	۴۲/۹۱	۶۲/۸۹	۱۹/۹۸	۰/۹۹۱۱
7.1.09	۲۵۶	۳۶/۱۳	۶۰/۴۱	۲۴/۲۸	۰/۹۷۸۴
7.1.10	۲۵۶	۴۱/۹۷	۶۵/۳۵	۲۳/۳۸	۰/۹۹
7.2.01	۱۲۸	۴۲/۸۸	۶۶/۸۵	۲۳/۹۷	۰/۹۸۷۲
Boat	۲۵۶	۴۰/۰۲	۵۹/۹۲	۱۹/۹	۰/۹۷۱۲
Elaine	۲۵۶	۳۸/۹۹	۵۵/۹۴	۱۶/۹۵	۰/۹۷۳۶
Gray21	۲۵۶	۳۵/۸۶	۵۲/۳۲	۱۶/۴۶	۰/۹۷۵۵
House	۲۵۶	۸۶/۷۸	۸۹/۴۸	۲/۷	۰/۹۵۶۴
Number	۲۵۶	۲۳/۵۶	۴۶/۳۲	۲۲/۷۶	۰/۸۹۷۷
Rulers	۶۴	۳۴/۲۴	۴۲/۰۷	۷/۸۳	۰/۹۸۱۰
testpat	۲۵۶	۴۳/۷۶	۶۰/۷۶	۱۷	۰/۹۹۹۹

جدول پ-۳: زمان پردازش هر یک از روش‌های پیشنهادی بر روی تصاویر پایگاه داده USD.

All data base	Image Dimension	Time method 1 (second)	Time method 2 (second)	Method 3	
				Block Dimension	Time (second)
4.1.01	۲۵۶×۲۵۶	۱/۹۱	۴/۳۶	۸	۷/۱۷۸
				۱۶	۲/۳۷
				۳۲	۲/۰۵
				۶۴	۱/۵۷
				۱۲۸	۱/۵۹
				۲۵۶	۱/۵۰
4.1.02	۲۵۶×۲۵۶	۱/۱۱	۲/۱۵۳	۸	۶/۷۵
				۱۶	۳/۰۳
				۳۲	۲/۱۵
				۶۴	۱/۸۵
				۱۲۸	۱/۷۵
				۲۵۶	۱/۴۳
4.1.03	۲۵۶×۲۵۶	۱/۱۳	۲/۰۴۶	۸	۷/۵۸
				۱۶	۲/۵۴
				۳۲	۲/۶۳
				۶۴	۱/۳۴
				۱۲۸	۱/۲۸
				۲۵۶	۱/۱۵
4.1.04	۲۵۶×۲۵۶	۱/۱۰	۲/۰۳	۸	۶/۹
				۱۶	۲/۴۵
				۳۲	۲/۳۸
				۶۴	۱/۳۲
				۱۲۸	۱/۱۸
				۲۵۶	۱/۰۷
4.1.05	۲۵۶×۲۵۶	۱/۶۵	۲/۰۴	۸	۷/۵۲
				۱۶	۲/۵۴
				۳۲	۲/۱۲
				۶۴	۱/۶۸
				۱۲۸	۱/۶۲
				۲۵۶	۱/۴۵
4.1.06	۲۵۶×۲۵۶	۲/۷۵	۲/۱۳	۸	۶/۵۸
				۱۶	۲/۴۷
				۳۲	۲/۳۵
				۶۴	۱/۶۸
				۱۲۸	۱/۲۳
				۲۵۶	۱/۱۸
				۸	۶/۵۴
				۱۶	۳/۱۳
				۳۲	۳/۰۳

All data base	Image dimension	Time method 1(second)	Time method2 (second)	method 3	
				Block dimension	Time (second)
4.1.07	256x256	1/13	2/34	64	1/98
				128	1/54
				256	1/78
4.1.08	256x256	1/0.66	2/0.6	8	7/18
				16	2/98
				32	2/65
				64	1/67
				128	1/54
4.2.01	512x512	3/13	3/69	256	1/43
				8	47/50
				16	9/20
				32	4/62
				64	4/0.7
4.2.02	512x512	4/23	3/87	128	3/61
				256	3/95
				8	45/28
				16	9/85
				32	4/87
4.2.03	512x512	3/92	3/68	64	4/35
				128	3/98
				256	3/42
				8	48/52
				16	10/0.2
4.2.04	512x512	4/0.6	3/63	32	4/32
				64	4/34
				128	3/23
				256	3/15
				8	46/27
4.2.05	512x512	3/98	3/92	16	9/97
				32	4/54
				64	3/95
				128	3/13
				256	3/15
4.2.06	512x512	3/97	3/74	8	44/28
				16	9/14
				32	4/54
				64	4/0.6
				128	3/85
4.2.06	512x512	3/97	3/74	256	3/65
				8	43/74
				16	11/25
				32	5/0.3
				64	4/99
4.2.06	512x512	3/97	3/74	128	3/87

All data base	Image dimension	Time method 1(second)	Time method2 (second)	Time method 3	
				Block dimension	Time (second)
				256	3/41
4.2.07	512x512	3/93	3/91	8	46/58
				16	10/19
				32	4/89
				64	3/74
				128	3/63
				256	3/43
5.1.09	256x256	1/54	2/12	8	7/91
				16	2/45
				32	2/53
				64	2/64
				128	1/95
				256	1/73
5.1.10	256x256	2/02	1/94	8	7/97
				16	2/67
				32	2/01
				64	1/19
				128	1/09
				256	0/9
5.1.11	256x256	1/68	2/05	8	7/17
				16	2/53
				32	2/09
				64	1/75
				128	1/45
				256	1/22
5.1.12	256x256	1/42	2/05	8	8/01
				16	2/45
				32	2/36
				64	1/95
				128	1/75
				256	1/11
5.1.13	256x256	1/23	2/09	8	7/83
				16	2/55
				32	2/32
				64	1/95
				128	1/63
				256	1/24
5.1.14	256x256	1/54	2/02	8	7/17
				16	2/43
				32	2/06
				64	1/59
				128	1/52
				256	1/49
				8	46/49

All data base	Image dimension	Time method 1(second)	Time method2 (second)	Time method 3	
				Block dimension	Time (second)
5.2.08	512×512	3/28	3/76	16	9/85
				32	4/62
				64	4/07
				128	3/53
				256	3/98
5.2.09	512×512	4/045	3/72	8	47/26
				16	9/97
				32	4/81
				64	4/64
				128	3/53
5.2.10	512×512	3/96	3/95	256	3/43
				8	46/67
				16	8/93
				32	4/56
				64	4/31
5.3.01	1024×1024	42/71	18/45	128	3/53
				256	3/42
				8	511/41
				16	142/46
				32	26/75
5.3.02	1024×1024	42/77	16/83	64	15/88
				128	9/71
				256	8/58
				8	509/34
				16	144/78
7.1.01	512×512	3/92	3/84	32	24/75
				64	16/11
				128	9/98
				256	8/23
				8	45/29
7.1.02	512×512	3/82	4/14	16	9/23
				32	4/78
				64	4/17
				128	3/41
				256	3/36
	512×512			8	46/74
				16	9/31
				32	4/79
				64	4/17
				128	3/11
				256	3/63
				8	46/21
				16	9/99
				32	4/67

All data base	Image dimension	Time method 1(second)	Time method 2 (second)	Time method 3	
				Block Dimension	Time (second)
7.1.03	512×512	3/88	3/64	64	4/0.4
				128	3/51
				256	3/48
7.1.04	512×512	3/80	3/97	8	46/23
				16	9/68
				32	4/98
				64	4/0.9
				128	3/46
				256	3/41
7.1.05	512×512	3/81	3/89	8	46/29
				16	9/57
				32	4/77
				64	4/25
				128	4/0.9
				256	3/25
7.1.06	512×512	3/81	4/32	8	41/96
				16	9/54
				32	4/36
				64	4/18
				128	3/96
				256	3/23
7.1.07	512×512	4/52	3/68	8	47/29
				16	9/89
				32	4/25
				64	4/16
				128	4/0.9
				256	3/24
7.1.08	512×512	3/85	3/86	8	47/29
				16	9/11
				32	4/78
				64	4/68
				128	4/12
				256	3/14
7.1.09	512×512	4/45	3/72	8	46/51
				16	9/41
				32	4/78
				64	4/34
				128	4/19
				256	3/88
7.1.10	512×512	3/96	4/0.9	8	46/11
				16	9/29
				32	4/91
				64	4/76
				128	4/18
				256	3/95
7.2.01				8	50/181
				16	141/26
				32	25/95

All data base	Image Dimension	Time method 1 (second)	Time method 2 (second)	Method 3	
				Block Dimension	Time (second)
	1.24x1.24	42/84	16/34	64	14/38
				128	1.0/54
				256	6/98
Boat	512x512	3/9.0	3/79	8	46/97
				16	9/64
				32	4/32
				64	4/11
				128	3/98
				256	3/86
Elaine	512x512	3/86	3/9.0	8	46/79
				16	9/45
				32	4/11
				64	4/0.1
				128	3/74
				256	3/71
gray21	512x512	3/83	3/99	8	46/97
				16	9/85
				32	4/23
				64	4/13
				128	3/98
				256	3/55
House	512x512	3/91	3/83	8	47/21
				16	9/98
				32	4/84
				64	4/36
				128	4/12
				256	3/8.0
Numbers	512x512	3/87	3/69	8	46/85
				16	9/65
				32	4/24
				64	4/12
				128	4/0.8
				256	3/41
Ruler	512x512	3/89	4/0.8	8	46/93
				16	9/28
				32	4/84
				64	4/65
				128	4/47
				256	3/99
Testpat	1.24x1.24	43/8	18/73	8	5.8/41
				16	143/67
				32	26/44
				64	14/98
				128	9/32
				256	7/52

مراجع

- [1] Gonzalez R. C., Woods R. E., (2007), "*Digital Image Processing*", Vol. 8, 3rd Edition, pp. 409-518.
- [2] Sayood K., (2005), "*Introduction to Data Compression*", 3rd Edition, California.
- [3] Singh V., (2013), "A Brief Introduction on Image Compression Techniques and Standards", *International Journal of Technology and Research Advances*, Vol. 3, issue 2, pp.19-23,
- [4] Yang Ch. and Bourbakis S., (2005), "An Overview of Lossless Digital Image Compression Techniques", *Circuits & Systems*, Vol. 2, pp 1099-1102.
- [5] Guy Blelloch E., (2013), "Introduction to Data Compression ", *Computer Science Department Carnegie Mellon University*, Vol.2, pp. 30-54.
- [6] Neil A. P., (2001), PhD thesis, " Extending Lossless Image Compression", University of Cambridge.
- [7] Kikuchi H. , Funahashi K. and Muramatsu S., (2009), "Simple bit-plane coding for lossless image compression and extended functionalities", *Picture Coding Symposium (PCS)*, Vol.2, pp. 23-27.
- [8] Podlasov A. , Fränti P. , (2006), "Lossless image compression via bit-plane separation and multi-layer context tree modeling", *Journal of Electronic Imaging*, Vol. 15, pp.1-3.
- [9] Golomb S. W., (1996), "Run-Length Encodings", *IEEE Trans. on Information Theory*, Vol. 12, pp. 399-401.
- [10] Setia V. and Kumar V., (2012), "Coding of DWT Coefficients using Run-length coding and Huffman Coding for the Purpose of Color Image Compression", *IJCA*, Vol. 27, No. 6, pp. 696-699.
- [11] Rubaiyat Hasan Md. , (2011), "Data Compression using Huffman based LZW Encoding Technique", *International Journal of Scientific & Engineering Research*, Vol 2, Issue 11, pp. 32-48.
- [12] Wikipedia Web page, (2014), Lampels Ziv welch, Available from: http://en.wikipedia.org/wiki/Lampels_ziv_welch , [Accessed 31 Aug 2014].
- [13] Nelson M., (1989), "LZW Data Compression", *Dr. Dobb's Journal*, Vol.14, No.10, pp. 29-36.
- [14] Saravanan C. and Ponalagusamy R., , (2011), "Lossless Grey-scale Image Compression using Source Symbols Reduction and Huffman Coding" , *International Journal of Image Processing*, Vol. 3, No.5, pp. 221-226.
- [15] Sharmap M., (2010), "Compression Using Huffman Coding", *International Journal of Computer Science and Network Security (IJCSNS)*, Vol.10, No.5, pp.133-141.
- [16] Wikipedia Web page, (2011), Huffman Coding, Available from: http://en.wikipedia.org/wiki/huffman_coding , [Accessed 31 Aug 2014].

- [17] Monika S. and Pragati Kapoor P., (2007), "A Huffman based LZW Approach to Perform Image Compression", *International Conference on Electrical and Electronics Engineering (ICEEE)*, Vol.9 ,pp.160-164.
- [18] Sindhu M., Rajkamal R.,(2009),"Images and Its Compression Techniques A Review", *International Journal of Recent Trends in Engineering*, Vol 2, No. 4, pp.71-75.
- [19] Ahmed N., Natarajan T. and Rao K. R., (1974), " Discrete cosine transform", *IEEE Trans Computers*, Vol. 23, pp. 90.93.
- [20] J. lee, N. Irwin, M. J ,(2006), "Inverse Discrete Cosine Transform Architecture Exploiting Sparseness and Symmetry Properties", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 16, pp. 655-662.
- [21] Gersho A., Gray M. ,(1992), "*Vector quantization and Signal Compression*", Kluwer Academic Publishers, Boston, pp. 291-302.
- [22] Yang Ch and Zhang S.,(1997), "A New Efficient Image Compression Technique with Index-Matching Vector Quantization", *IEEE Trans. on Consumer Electronics*, Vol. 43, Issue 2, pp. 173- 182.
- [23] Jacquin A.E,(1992), "Image coding based on a fractal theory of iterated contractive image transformations", *IEEE Trans.on Image Processing*, Vol .1, pp. 18-30.
- [24] Jacguine A.E, (1993), "Fractal Image Coding",*Proceeding of IEEE*, Vol. 81, pp.1451-1465.
- [25] Barnsley M.F, and Demko S.,(1985),"Iterated function systems and the global construction of fractals", *Proc. Roy. Soc. Lond.*,Vol. 399, pp. 243-275.
- [26]Yang C., Lin J., and Tsai W.,(1997),"Color Image Compression by Momentpreserving and Block Truncation Coding Techniques", *IEEE Trans.Commun.*, Vol.45, No.12, pp.1513-1516.
- [27] E.G. Delpand and O.R Mitchell,"Image compression using bluck truncation coding", *IEEE Trans.Commun.*, Vol. 27, pp. 1335-1342.
- [28] The JPEG web page, (2014), JPEG Coding, Available from: <http://www.jpeg.org/jpeg>, [Accessed 28 Jul 2014].
- [29] Wallace G. K., (1991), "The jpeg still picture compression standard", *Communications of the ACM*, Vol. 34, Issue. 4, pp. 30-44.
- [30] Maan A. J., (2013), "An Introduction to JPEG Image Compression algorithm", *International Journal of Electrical, Electronics and Data Communication*,Vol. 1, pp.44-46.
- [31] Ansari R. and Memon N.,(2000)," *The JPEG lossless compression standards*", *Handbook of image and video processing*, pp. 599-607.
- [32] Wikipedia web page, (2014), Lossless JPEG Coding, Available from: http://en.wikipedia.org/wiki/Lossless_JPEG, [Accessed 1 Agu 2014].
- [33] Weinberger M.J., Seroussi G., Sapiro G., (2000) ,"The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS", *IEEE Transactions on Image Processing*, Vol. 9, issue. 8, pp. 1309-1324.

- [34] Weinberger M. J., Seroussi, G. and Sapiro G.,(1996), “LOCO-I: A low complexity, context-based, lossless image compression algorithm”, Data Compression Conference, pp. 140–149.
- [35] Rabbani M. and Santa Cruz D., (2001), “The JPEG2000 still-image compression standard”, Signal Processing Magazine, Vol.18, Issue. 5, pp. 36-58.
- [36] Marcellin M. W., Gormish M. J., Bilgin A. and Boliek M. P., (2000), “An overview of JPEG-2000”, IEEE Data Compression Conference (DCC), pp. 523-541.
- [37] Taubman D.S., Marcellin M.W., (2002), “JPEG2000: Standard for interactive imaging”, *Proceedings of the IEEE*, Vol. 90, issue 8, pp. 1336-1357.
- [38] Marcellin M. W., Gormish M. J., A. Bilgin, and M. P. Boliek, (2000), “An Overview of JPEG2000”, Data Compression Conference, pp. 523–544.
- [39] Hoon Yap M., Bister M., Tat Ewe H., (2003), “Gaussian Blurring-Deblurring for Improved Image Compression”, VIIth Digital Image Computing: Techniques and Applications, (DICTA), pp. 10-12.
- [40] Pinho A. J., (2002), “An online preprocessing technique for improving the lossless compression of images with sparse histograms”, *IEEE Signal Processing Letters*, vol. 9, No. 1, pp. 5-7.
- [41] Ferreira P. J. and Pinho A. J., (2002), “Why does histogram packing improve lossless compression rates?”, *IEEE Signal Processing Letters*, Vol. 9, No. 8, pp. 259-261.
- [42] J.Pinho A., and J. R. Neves A.,(2003), “Block-based histogram packing of color-quantized images”, International conference on Multimedia and Expo. (ICME) ,Vol. 1, pp.341-344.
- [43] J.Pinho A., and J. R. Neves A.,(2004), “Variable Size Block-Based Histogram Packing For Lossless Coding of Color-Quantized Images”, Proceedings of the Fourth International conference Visualization, Imaging, And Image Processing, Vol. 2, pp. 738-741.
- [44] Nasr-Esfahani E., Samavi S., Karimi N. and Shirani S., (2007), “Near- Lossless Image Compression Based on Maximization of Run Length Sequences”, ICIP, Vol.4, pp. 177-180.
- [45] Avcibas A., Memon V., Sankur D. and Sayood Kh.,(2002), “A Progressive Lossless/Near Lossless Image Compression Algorithm”, *IEEE Signal Processing Letters*, Vol. 9, No. 10, pp. 312-314.
- [46] Nasr-Esfahani E., Samavi S. and Karimi N., (2008), “Near lossless Image Compression By Local Packing of Histogram”, ICASSP, pp.1197-1200.
- [47] Forney G.D., (1973), “The Viterbi algorithm”, *Proc. IEEE*, Vol .61, pp. 268–278.
- [48] Kim S.,Hwang D.,Yoo G.,Soo You K. and Kwak S., (2005), “A preprocessing algorithm for efficient lossless compression of gray scale image”, ICCAS, pp. 73-77.
- [49] B. Park¹; Y. R. Chen² ,(2001), “Co-occurrence Matrix Texture Features of Multi-spectral Images on Poultry Carcasses “, Automation and Emerging Technologies, Vol. 78, pp. 127-139.
- [50] karthikeyan, B., vaithianathan, V., Venkatraman B. and Menaka M.,(2014), “A performance analysis of different Scanning Path on lossless Image compression for Radiographic Welding Image”, *Journal of Scientific & industrial research*, Vol. 73, pp. 214-218.

- [51] Jalumuri N. R., (2004), Master's thesis, "A Study of Scanning Paths for BWT Based Image Compression., West Virginia University.
- [52] Elfitrin S., PhD thesis, (2011), "Lossless and nearly-lossless image compression based on combinatorial", University of Cambridge.
- [53] Manzini G., (1999), "An analysis of the Burrows-Wheeler transform", *Journal of the ACM*, Vol.48, pp. 262-272.
- [54] Syahrul E., Dubois J., Vajnovszki V., Saidani T. and Atri M, (2008), "Lossless image compression using Burrows Wheeler Transform (methods and techniques)", *Signal Image technology and internet –based systems (SITIS)*, Vol.8, pp. 338–343.
- [55] Fenwick. P. M. , (2007), "Burrows–Wheeler compression: Principles and reflections", *Theoretical computer science conference*, Vol. 387, No. 3, pp. 200–219.
- [56] Herrigel A., Baumgarther N., (1999), "a stochastic approach to content adaptive digital image watermarking", *lecture note in computer science, proceeding of the third international workshop on information hiding*, Vol. 1768, pp. 211-236
- [57] Douak F., Benzid R. and Benoudjit N., (2010), "Color image compression algorithm based on the DCT transform combined to an adaptive block scanning", *International Journal of Electronics and Communications*, Vol. 11, pp. 16-26.
- [58] Walaa M., Elhafiez A., (2012), "New Approach for Color Image Compression", *International Journal of Computer Science and Telecommunications (IJCST)*, Vol. 3, Issue 4, pp. 14-19.
- [59] Boucetta A., Melkemi K., (2012), "Dwt based-Approach for Color Image Compression Using Genetic Algorithm", 5th international Conference (ICISP), Vol. 7340, pp. 476-484.
- [60] University Of Southern California web site, (2014), USD Database, Available from : <http://sipi.usc.edu/database/?volume=misc>, [Accessed 31 Aug 2014].
- [61] Kodak Lossless True Color Image Suite web site, (2014), Kodak Database, Available from: <http://r0k.us/graphics/kodak>, [Accessed 31 Aug 2014].
- [62] Grgic S. , Mark M., (2004), "reliability of objective picture quality measures", *Journal of Electrical Engineering*, Vol. 55, No. 1-2, pp. 3-10.
- [63] Mark M., Grgic S. , Grgic M., (2003) , "Picture quality_measures in image compression systems", *EUROCON conference* , Vol. 1, pp. 233-236.
- [64] Avacibas L., Sankur B. and Sayood K., (2002), "Statistical evaluation of image quality measures", *Journal of Electronic Imaging* , Vol.11, pp. 206-223.
- [65] Wang Z. , Bovik A. C., Sheikh H. R. and Simoncelli E. P., (2004), "Image quality assessment From error visibility to structure similarity", *IEEE Transactions on Image Processing*, Vol.13, pp. 600-612.
- [66] Wang Z. and Bovik A.C., (2002), "A universal image quality index", *IEEE Signal Processing Letters* , Vol.9, pp. 81-84.
- [67] Whitley D., (2003), "A genetic algorithm tutorial" , *Statistics and computing*, Vol. 4, pp. 65-85.
- [68] Tang, K. S., Man K. F., Kwong S. and He Q., (1996), "Genetic algorithms and their applications", *IEEE Signal Processing Magazine*, Vol. 13, pp. 22-37.

Abstract

Currently, due to the spread and high volume of digital information, image compression is presented as a basic requirement in the archive and sending systems. Therefore, a lot of methods have been established to increase the compression rate of image. However, increasing compression rate of image and maintain of the retrieved image is still regarded as a necessity.

Each of the available compression method attempts to identify the redundancy and consequently to compress it based on one or more features. Accordingly, each of the image compression algorithms doesn't present the same compression rate for different images. Compression rates of these algorithms may depend on some factors such as the variety of the grey levels, smoothness of the image and the uniformity of the adjacent pixels in the image. Accordingly, some solutions are presented in this study which is applied on the image as a pre-processing stage so that they reinforce the effective parameters in the image toward the compression.

The first processing is lossless and formed in two ways. In the first step image is divided into non-overlapping blocks and the blocks are arranged based on the coefficient increase (decrease) of DC. In the second step, eight reversible geometric transforms are shaped on the resulted image of the first step. Among the resulted images, an image is selected which leads to the best result for compression. This process reduces the scope of the changes in the results of the JPEG calculations and improves the compression rate of it.

In the second method, the Grey levels that are quantitatively closer are categorized in one group. Then, a grey level is considered as a representative for each group. The grey level value of each pixel of the original image is replaced with the related representative value of its group. That is why this value has been chosen so that the minimum destruction of the image occurs. In the next stage, the histogram of the pre-processed image is packed. This pre-process as well as the preservation of the image quality, improves the compression rate of lossless JPGE.

Third method is same with second method with this difference that image is blocked in fixed sizes and value of the parameter is selected based on the human vision system for each block. This pre-processing improves the compression rate of JPEG2000 and Lossless JPEG algorithms.

In the fourth method, using Genetic algorithm, a suitable color space is created for compressing each image separately. This pre-process on compression algorithm has improved performance of CIC algorithm.

Keyword:

Image compression, Redundancy, Processing, improving the compression rate, Histogram image, Lossless JPEG, Human visual system, JPEG2000, color Space.



University of Shahrood

Faculty of Computer Engineering and Information Technology

**Using suitable pre-processing approaches for improving
performance of image compression techniques**

Zahra Gheisi

Supervisor:

Prof. Hamid Hassanpour

SEP 2014