

## بهبود عملکرد کدگذار پیشگو با تغییر چیدمان تصویر به کمک الگوریتم ژنتیک

سکینه اسدی امیری<sup>۱</sup>، حمید حسن پور<sup>۲</sup>

### چکیده

روش‌های فشرده‌سازی تصویر را می‌توان به دو دسته بااتلاف و بی‌اتلاف تقسیم‌بندی نمود. کدگذار پیشگو مبنای بسیاری از روش‌های فشرده‌سازی بی‌اتلاف تصویر است. این کدگذار با توجه به مقدار پیکسل‌های همسایه، مقداری را برای هر پیکسل از تصویر پیشگویی می‌نماید. تفاضل مقدار واقعی هر پیکسل از مقدار پیشگویی شده، مقدار خطای تلقی می‌شود و این مقادیر خطا کد می‌گردند. در این مقاله، روش پیش‌پردازشی پیشنهاد شده است که چیدمان تصویر را طوری تغییر می‌دهد تا مقادیر پیکسل‌های همسایه، همبستگی بیشتری با هم داشته باشند. با افزایش همبستگی بین پیکسل‌های همسایه، کدگذار پیشگو می‌تواند مقدار دقیق‌تری را برای هر پیکسل پیشگویی نماید، در نتیجه آنتروپی در تصویر خطا کاهش می‌یابد. طبق نظریه اطلاعات هر چه آنتروپی تصویر کمتر باشد، قابلیت کدگذار آنتروپی در فشرده‌سازی آن افزایش می‌یابد. در روش پیشنهادی به کمک الگوریتم ژنتیک، تبدیلات هندسی چرخش و انعکاس مناسبی بر روی هر بلوک از تصویر اعمال می‌شود تا چیدمان تصویر طوری تغییر یابد که همبستگی پیکسل‌های مجاور تصویر افزایش یابد. در این مقاله دو روش فشرده‌سازی JPEG بی‌اتلاف و CALIC که مبتنی بر کدگذار پیشگو هستند مورد بررسی قرار گرفت. نتایج ارزیابی روش پیشنهادی بر روی تصاویر متعدد نشان می‌دهد پیش‌پردازش پیشنهادی نرخ فشرده‌سازی این دو روش را بهبود می‌بخشد.

### کلید واژه‌ها

فشرده‌سازی تصویر، JPEG بی‌اتلاف، CALIC، تبدیلات هندسی، کدگذار پیشگو.

می‌شود. این افزونگی سبب می‌شود که انرژی تصویر در فرکانس‌های پایین آن تجمع یابند. برای حذف یا کاهش همبستگی مکانی از عمل نگاشت استفاده می‌شود، بدین گونه که تصویر از حوزه مکان به حوزه دیگر منتقل می‌گردد [۲]. افزونگی آماری به معنای تنوع در فراوانی سطوح خاکستری موجود در تصویر می‌باشد. برای حذف افزونگی آماری می‌توان از کدگذارهای آنتروپی همچون هافمن، حسابی و گلمب<sup>۱</sup> استفاده نمود. در این کدگذارها به مقادیر خاکستری با فراوانی زیاد کد بیتی کوتاه‌تر و به سطوح خاکستری با فراوانی کم، کد بیتی طولانی‌تری تخصیص می‌یابد [۳].

### ۱ مقدمه

به دلیل رشد سریع داده‌های موجود، مسأله فشرده‌سازی اهمیت بسزایی دارد. هدف از فشرده‌سازی، کم کردن حجم داده با توجه به کاهش (حذف) افزونگی‌های موجود در آن است. مبنای اصلی روش‌های فشرده‌سازی براساس سه نوع افزونگی مکانی، افزونگی آماری و افزونگی دیداری می‌باشد [۱]. همبستگی مقادیر پیکسل‌های مجاور تصویر منجر به افزونگی مکانی در تصویر

این مقاله در اسفندماه ۱۳۹۳ دریافت، در مهرماه ۱۳۹۴ بازنگری و در آبانماه همان سال پذیرفته شد.

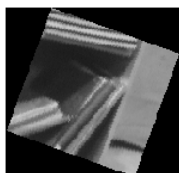
<sup>۱</sup> دانشجوی دکتری، دانشکده مهندسی کامپیوتر، دانشگاه شاهرود

رایانامه: [asadi\\_amiri@yahoo.com](mailto:asadi_amiri@yahoo.com)

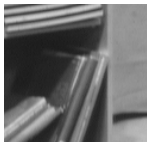
<sup>۲</sup> دانشکده مهندسی کامپیوتر، دانشگاه شاهرود

رایانامه: [h.hassanpour@shahroodut.ac.ir](mailto:h.hassanpour@shahroodut.ac.ir)

بر کدگذار پیشگو همچون JPEG بی‌اتلاف و CALIC بهبود می‌یابد. در روش پیشنهادی ابتدا تصویر به بلوک‌هایی تقسیم می‌شود، سپس به هر یک از بلوک‌های تصویر، تبدیلات هندسی چرخش و انعکاس مناسبی اعمال می‌گردد تا همبستگی پیکسل‌های مجاور تصویر افزایش یابد. برای انتخاب تبدیل هندسی مناسب برای بلوک‌های تصویر از الگوریتم ژنتیک استفاده شده است. قابل ذکر است تبدیلات هندسی اعمال شده به تصویر، چرخش در زاویه‌های ۰، ۹۰، ۱۸۰ و ۲۷۰ درجه و انعکاس در جهت‌های محور عمودی (y)، محور افقی (x)، محور  $y=x$  و محور  $y=-x$  می‌باشند. همچنین چرخش تصویر در زاویه‌های دیگر منجر به بزرگتر شدن اندازه تصویر می‌شود. از اینرو برای فشرده‌سازی تصویر، پیش‌پردازش مناسبی نمی‌باشند. شکل ۱ نمونه‌ای از چرخش یک بلوک از تصویر در اندازه  $64 \times 64$  را در زاویه ۲۰ درجه نشان می‌دهد. همان‌طور که در این شکل مشخص است، این چرخش منجر به بزرگتر شدن اندازه بلوک تصویر شده است.



الف) یک بلوک از تصویر، اندازه  $64 \times 64$   
ب) بلوک چرخش‌یافته، اندازه  $71 \times 71$



شکل ۱ نمونه‌ای از چرخش یک بلوک از تصویر در زاویه ۲۰ درجه.

در بخش بعدی مروری بر کارهای پیشین در زمینه فشرده‌سازی تصویر انجام می‌شود. در بخش سوم دو روش فشرده‌سازی JPEG بی‌اتلاف و CALIC شرح داده می‌شود. روش پیشنهادی در بخش چهارم ارائه شده است. نتایج این مقاله در بخش پنجم و نتیجه‌گیری نیز در بخش ششم مطرح شده‌اند.

## ۲ مروری بر کارهای پیشین

همان‌طور که اشاره شد در بسیاری از موارد نیاز است فشرده‌سازی به صورت بی‌اتلاف بر روی تصویر انجام گیرد. از اینرو روش‌های بسیاری در این زمینه مطرح شده‌اند. همه این روش‌ها مبتنی بر دو دیدگاه می‌باشند. دیدگاه اول، ارائه یک روش فشرده‌سازی است و دیدگاه دیگر، بهبود روش‌های فشرده‌سازی موجود با انجام پیش‌پردازش بر روی تصویر اولیه می‌باشد. در دیدگاه دوم، تصویر اولیه طوری پیش‌پردازش می‌گردد تا با این پیش‌پردازش، پارامتر تأثیرگذار در روش فشرده‌سازی مورد نظر تقویت گردد. در واقع فرآیند پیش‌پردازش مطابق شکل ۲ الف) قبل از فرآیند کدگذاری بر روی تصویر انجام می‌شود. برای بازیابی تصویر نیاز است که مطابق شکل ۲ ب) پس‌پردازشی به تصویر کدگشایی شده اعمال گردد.

افزونگی دیداری به این معنا است که همه نواحی تصویر از نظر سیستم بینایی انسان به یک میزان اهمیت ندارند و برخی از اطلاعات تصویر در سیستم بینایی انسان نادیده گرفته می‌شوند. از اینرو این اطلاعات افزونه را می‌توان از تصویر حذف نمود، بدون اینکه حذف این اطلاعات برای چشم انسان چندان قابل درک باشد. حذف افزونگی دیداری در روش‌های فشرده‌سازی بااتلاف انجام می‌شود [۴].

در بسیاری از کاربردها همچون تصاویر پزشکی [۵] و فایل‌های اجرایی نیاز است که فشرده‌سازی به صورت بی‌اتلاف انجام گیرد. بر این اساس روش‌های فشرده‌سازی بدون اتلاف زیادی همچون [۶] LZW، [۷] Packbits، [۸] بی‌اتلاف [۸]، JPEG 2000 بی‌اتلاف [۹] و CALIC [۱۰] مطرح شده‌اند. هر یک از این روش‌ها بر روی یک تصویر، نرخ فشرده‌سازی متفاوتی را ارائه می‌دهند. در واقع هر یک از این روش‌های فشرده‌سازی منطبق بر الگوریتم فشرده‌سازی خاصی می‌باشند. از اینرو می‌توان با شناسایی عوامل تأثیرگذار در عملکرد یک روش فشرده‌سازی و تقویت آنها در مرحله پیش‌پردازش، نرخ فشرده‌سازی آن روش را بهبود بخشید. برای نمونه اگر بتوان پیش‌پردازشی بر روی تصویر اعمال نمود تا آنتروپی تصویر حاصل کاهش یابد، نرخ فشرده‌سازی کدگذار آنتروپی در فشرده‌سازی آن افزایش می‌یابد.

روش‌های فشرده‌سازی JPEG بی‌اتلاف و CALIC دو روش فشرده‌سازی بدون اتلاف مبتنی بر کدگذار پیشگو هستند. در کدگذار پیشگو مقدار هر پیکسل از تصویر براساس پیکسل‌های همسایه آن پیشگویی می‌شود. سپس تفاضل مقدار واقعی پیکسل از مقدار پیشگویی شده به عنوان مقدار خطا برای آن پیکسل در نظر گرفته می‌شود و این مقادیر توسط کدگذار آنتروپی کد می‌شوند [۱۱]. اگر کدگذار پیشگو بتواند مقدار نسبتاً دقیقی برای هر پیکسل پیشگویی نماید، تصویر حاصل از مقادیر خطا حاوی مقادیری اندک و معدود است که آنتروپی این تصویر بسیار کمتر از آنتروپی تصویر اولیه است. از اینرو کدگذار آنتروپی با نرخ فشرده‌سازی بیشتری می‌تواند این تصویر خطا را کد نماید. بر این اساس روش‌های کدگذار پیشگوی متعددی ارائه شده‌اند تا بتوانند مقدار دقیق‌تری را برای هر پیکسل از تصویر پیشگویی نمایند.

در این مقاله دیدگاه دیگری برای بهبود عملکرد کدگذار پیشگو ارائه شده است. در این دیدگاه، به جای ارائه یک روش جدید کدگذار پیشگو، چیدمان تصویر طوری تغییر می‌یابد تا همبستگی پیکسل‌های مجاور تصویر افزایش یابد. بر این اساس کدگذار پیشگو قادر است مقدار دقیق‌تری را برای هر پیکسل از تصویر پیشگویی نماید. از اینرو با این پیش‌پردازشی که بر روی تصویر انجام می‌شود، عملکرد روش‌های فشرده‌سازی بدون اتلاف مبتنی

<sup>۱</sup> Lampel-Ziv-Welch

<sup>۲</sup> Joint Photographic Experts Group

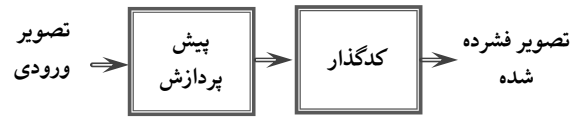
<sup>۳</sup> Context based Adaptive Lossless Image Codec

وایتری<sup>۱</sup> به هیستوگرامی با آنتروپی کمتر تبدیل می‌شود. سپس تصویر متناظر با این هیستوگرام ایجاد می‌گردد. این تصویر پیش‌پردازش شده، قابلیت فشرده‌سازی JPEG را بر روی برخی از تصاویر افزایش می‌دهد.

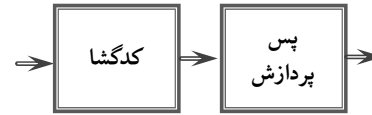
در [۱۷] یک روش فشرده‌سازی بی‌اتلاف با استفاده از افزونگی‌های مکانی بین پیکسل‌های تصویر پیشنهاد شده است. در این روش، تفاضل مقادیر پیکسل‌های ستون‌های فرد تصویر از مقادیر ستون‌های زوج مجاور محاسبه می‌گردد. سپس مقادیر تفاضل در یک ماتریس نگهداری شده و مقادیر ستون‌های فرد برای تکرار بعدی لحاظ می‌شوند. در این مرحله فرآیند تفاضل به صورت سطر به این مقادیر باقیمانده اعمال می‌شود. مشابه قبل مقادیر حاصل از تفاضل در ماتریسی نگهداری می‌شوند و سطرهای فرد برای تکرار بعدی لحاظ می‌شوند. این فرآیند به صورت تکراری بر روی سطرها و ستون‌ها اعمال می‌شود تا اندازه تصویر باقیمانده به یک مقدار آستانه از قبل تعریف شده برسد. سرانجام مقادیر تفاضل که آنتروپی آن از تصویر مرجع کمتر است با یک کدگذار آنتروپی کد می‌شوند. نرخ فشرده‌سازی روش پیشنهادی نزدیک به JPEG بی‌اتلاف است.

در [۱۸] یک کدگذار پیشگوی جدید ارائه شده است تا مقدار دقیق‌تری را برای نواحی لبه و بافت متراکم تصویر پیشگویی نماید. در این روش، ابتدا تصویر به دو زیر تصویر تقسیم می‌شود. زیر تصویر اول براساس سطرهای زوج تصویر و زیر تصویر دوم براساس سطرهای فرد تصویر به وجود می‌آید. سپس زیر تصویر اول براساس ستون‌های فرد به دو زیر تصویر جدید تقسیم می‌شود. به همین روال، زیر تصویر حاصل اول، براساس سطرهای زوج و فرد به دو زیر تصویر دیگر تقسیم می‌شود. این فرآیند تجزیه زیر تصویرها، به صورت سلسه مراتبی تا چند مرحله ادامه می‌یابد. همان‌طور که مشخص است، هر یک از دو زیر تصویر در هر مرحله شباهت زیادی به هم دارند، در نتیجه با توجه به اطلاعات یکی از این زیر تصویرها، می‌توان زیر تصویر دیگر را به صورت دقیق‌تر با کدگذار پیشگو، پیشگویی نمود. برای کدگذاری این تصویر، از مرحله آخر شروع نموده و زیر تصویر اول در این مرحله، با کدگذار پیشگو کد می‌گردد. از آنجایی که زیر تصویر دوم با زیر تصویر اول همان مرحله شباهت دارد، با توجه به نتایج کدگذار پیشگو برای زیر تصویر اول، پیشگویی دقیق‌تری برای زیر تصویر دوم آن مرحله انجام می‌شود. این فرآیند از انتها تا ابتدا انجام می‌شود تا کل تصویر با کدگذار پیشگو کدگذاری گردد. بدین‌گونه پیشگویی دقیق‌تری برای پیکسل‌ها انجام می‌شود. سپس نتایج حاصل با کدگذار حسابی کد می‌گردد.

در [۱۹] ابتدا تصویر به نواحی هموار و نواحی غیرهموار تقسیم می‌شود. سپس هر یک از دو ناحیه به صورت مجزا فشرده می‌گردند. برای هر بلوک ناحیه غیرهموار، شبیه‌ترین بلوک از میان



الف) اعمال پیش‌پردازش به کدگذار



ب) اعمال پس‌پردازش به کدگشا

شکل ۲ اعمال فرآیند پیش‌پردازش و پس‌پردازش برای بهبود عملکرد الگوریتم‌های فشرده‌سازی.

علی‌رغم کارایی نقش پیش‌پردازش در فشرده‌سازی تصویر، این موضوع کمتر مورد توجه قرار گرفته است. در [۱۲] روشی مبتنی بر ماتریس هم‌رخداد برای کاهش تنوع سطوح خاکستری تصویر ارائه شده است. این روش پیش‌پردازش، به عنوان یک روش بی‌اتلاف نرخ فشرده‌سازی روش هافمن را - اگر اندازه تصویر بزرگتر از  $256 \times 256$  باشد - می‌تواند بهبود دهد. در [۱۳] روش پیش‌پردازش بی‌اتلافی بر مبنای افزایش همبستگی پیکسل‌های تصویر ارائه شده است. در این روش، تصویر به بلوک‌های ثابت  $32 \times 32$  تقسیم می‌شود. سپس به ازای هر بلوک به کمک جدول نگاشت مناسب، مقادیر سطوح خاکستری تغییر می‌یابند تا دامنه تغییرات این سطوح کاهش یابند. نقص این روش در این است که از بلوک‌هایی با اندازه ثابت و کوچک برای پیش‌پردازش استفاده می‌شود. بر این اساس تعداد بلوک‌های ایجاد شده در تصویر زیاد خواهد شد. از آنجایی که به ازای هر بلوک، جدول نگاشت مربوط به آن نیز ذخیره می‌شود، فضای قابل توجهی صرف ذخیره‌سازی جدول نگاشت خواهد شد. نسخه بهبود یافته این روش در [۱۴] ارائه شده است که در آن، اندازه پنجره متغیر در نظر گرفته می‌شود. برای نواحی یکنواخت اندازه پنجره بزرگتر و برای نواحی با بافت متراکم، اندازه پنجره کوچکتر لحاظ می‌شود. اگر چه در این روش، تعداد بلوک‌های ایجاد شده برای هر تصویر کاهش می‌یابد ولی همچنان حجم قابل توجهی برای ذخیره‌سازی جدول نگاشت مصرف می‌شود.

در روش پیش‌پردازش مطرح شده در [۱۵] با توجه به پیکسل‌های هر سطر از تصویر، گرافی برای هر سطر ایجاد می‌شود که هدف آن یافتن یک مسیر بهینه در این گراف است به طوری که مقادیر پیکسل‌های مشابه در مجاورت هم در یک سطر قرار گیرند. در واقع با توجه به نگاشتی که بر روی هر سطر از تصویر انجام می‌گیرد، مقادیر خاکستری پیکسل‌ها طوری تغییر می‌یابند که پیکسل‌های بیشتری با مقادیر یکسان در مجاورت هم قرار گیرند. این تصویر پیش‌پردازش شده تا حدی قابلیت JPEG را بهبود می‌بخشد. در [۱۶] هیستوگرام تصویر اصلی به کمک الگوریتم

هستند و تنوع آنها نیز کم است، به تعداد بیت کمتری نسبت به تصویر اصلی برای ذخیره‌سازی نیاز دارند.

$$e = P - \hat{P} \quad (1)$$

در رابطه (۱)،  $P$  مقدار پیکسل جاری و  $\hat{P}$  مقدار پیشگویی شده برای این پیکسل می‌باشد. اگر مقادیر خطای پیشگویی شده بدون هیچ تغییری ذخیره شوند، فشردگی بی‌اتلاف است. در حالت بااتلاف مقادیر خطای چندی‌سازی می‌گردند و سپس کد می‌شوند. برای کدگشایی تصویر بسته به حالت پیشگویی انتخابی (جدول ۱)، نیاز است که مقادیر تعدادی از پیکسل‌های اولیه نیز به عنوان سربار این روش کد گردند. قابل ذکر است JPEG بی‌اتلاف از کدگذار پیشگو مطابق با حالت شش جدول ۱ استفاده می‌نماید. بسیاری از کدگذارهای پیشگو قادر به پیشگویی لبه‌ها و بافت‌های متراکم تصویر نمی‌باشند. از اینرو کدگذار پیشگویی کارآمد است که بتواند مقدار نسبتاً دقیقی را برای این پیکسل‌ها پیشگویی نماید.

شکل ۴ یک تصویر مرجع و تصویر خطای حاصل از کدگذار پیشگو مطابق با حالت شش جدول ۱ را به همراه هیستوگرام‌های مربوطه نشان می‌دهد. همان‌طور که از هیستوگرام مشخص است، تنوع مقادیر حاصل از کدگذار پیشگو بسیار کمتر از تنوع سطوح خاکستری موجود در تصویر است و این مقادیر اعدادی کوچک در محدوده صفر می‌باشند.

		M	M	M
	M			

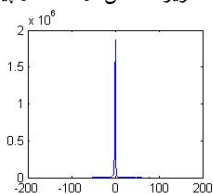
شکل ۳ نمایش چند پیکسل همسایه برای پیکسل  $P$ .

جدول ۱ نمایش چند حالت برای پیشگویی مقدار پیکسل  $P$ .

شماره حالت	پیشگویی کننده ( $\hat{P}$ )
۱	
۲	
۳	
۴	
۵	
۶	



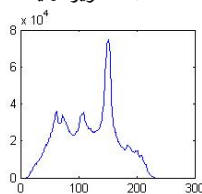
(ب) تصویر حاصل از کدگذار پیشگو



(ت) هیستوگرام تصویر (ب)



(الف) تصویر اولیه



(پ) هیستوگرام تصویر (الف)

شکل ۴ نمونه‌ای از عملکرد کدگذار پیشگو به همراه هیستوگرام.

بلوک‌های ناحیه غیرهمواری که کد شده‌اند یافت می‌شود و اختلاف این دو بلوک کد می‌گردد. برای فشردگی نواحی هموار نیز از روش CALIC استفاده شده است. روش فشردگی بی‌اتلاف پیشنهاد شده در [۲۰] از سه بخش اصلی تبدیل موجک، هادامارد و کدگذار آنتروپی تشکیل شده است. ابتدا تبدیل موجک به تصویر اعمال می‌شود. سپس برای حذف افزونگی بیشتر، هر یک از زیرباندهای موجک، به بلوک‌های  $4 \times 4$  تقسیم می‌شود و تبدیل هادامارد به هر یک از این بلوک‌ها اعمال می‌گردد تا همبستگی درون این بلوک‌ها حذف شود. سپس هر بلوک به صورت مجزا توسط یک روش کدگذار مبتنی بر آنتروپی کد می‌گردد. نرخ فشردگی روش مطرح شده به JPEG 2000 نزدیک است، ولی روش مطرح شده بسیار سریعتر از JPEG 2000 است.

در روش فشردگی بی‌اتلاف پیشنهاد شده در [۲۱] ابتدا تصویر به خطوطی تقسیم می‌گردند. سپس این خطوط براساس میزان شباهت آنها نسبت به هم به چند کلاس تقسیم شده و هر کلاس که در واقع حاوی پیکسل‌هایی با همبستگی زیاد است به کدگذار پیشگو اعمال می‌شوند. کدگذار پیشگو مقداری را برای هر پیکسل پیشگویی می‌نماید و تفاضل مقدار واقعی از مقدار پیشگویی شده محاسبه می‌شود. در پایان مقادیر خطای حاصل از کدگذار پیشگو با کدگذار آنتروپی کد می‌گردند.

### ۳ روش‌های فشردگی سازی JPEG بی‌اتلاف و CALIC

JPEG بی‌اتلاف و CALIC دو روش فشردگی بی‌اتلاف هستند که از دو فاز اصلی کدگذار پیشگو و کدگذار آنتروپی تشکیل شده‌اند. در ادامه هر یک از این دو کدگذار به اختصار شرح داده می‌شوند.

#### ۳-۱ کدگذار پیشگو

کدگذار پیشگو برای حذف یا کاهش افزونگی مکانی پیشنهاد شده است و می‌تواند به صورت بااتلاف یا بی‌اتلاف باشد [۱]. همان‌طور که اشاره شد، در این روش، مقدار هر پیکسل با توجه به همسایه‌های آن پیشگویی می‌شود و مقدار حاصل از تفاضل مقدار واقعی از مقدار پیشگویی شده که خطای پیشگویی نام دارد، محاسبه می‌گردد. برای تعیین مقدار پیشگویی برای پیکسل جاری  $P$  می‌توان از همسایگی‌های نشان داده شده در شکل ۳ استفاده نمود. روش‌های مختلفی برای پیشگویی مقدار پیکسل  $P$  مطرح شدند که در جدول ۱، چند مورد از پیشگویی‌ها نشان داده شده است [۲۲]. با توجه به رابطه (۱) برای هر پیکسل، مقدار خطایی ( $e$ ) به دست می‌آید که این مقادیر خطا توسط یک روش کدگذار آنتروپی کد می‌شوند. از آنجایی که مقادیر خطا اعداد کوچکی

است، کد بیته کوتاها تر و به نمادهایی که احتمال وقوع آنها کمتر است، کد بیته بزرگتری اختصاص داد. در واقع از کدهای با طول متغیر استفاده نمود. رابطه بالا بیانگر این است که امکان کد کردن بی اتلاف مقادیر شدت روشنایی تصویر با کمتر از  $En$  بیت بر پیکسل وجود ندارد. در واقع مقدار آنتروپی بیانگر یک حد پایین برای فشرده سازی بی اتلاف برای کدگذاری یک به یک پیکسل های مستقل آماری می باشد [۱]. آنتروپی هر یک از تصاویر شکل ۴ با توجه به رابطه بالا به ترتیب مقدار  $۷/۴۹۳$  و  $۳/۷۴$  برای شکل های ۴ الف) و ب) به دست آمد. یعنی حد پایین فشرده سازی با کدگذار آنتروپی برای هر یک از این تصاویر، برابر این مقادیر آنتروپی می باشد.

#### ۴ روش پیشنهادی

همان طور که اشاره شد، کدگذار پیشگو یکی از کدگذارهای موثر در بسیاری از روش های فشرده سازی بدون اتلاف تصویر است. هر چه این کدگذار بتواند مقدار دقیق تری را برای هر پیکسل از تصویر پیشگویی نماید، عملکرد کدگذار پیشگو و به طور کل عملکرد آن روش فشرده سازی بدون اتلاف بهبود می یابد. بر این اساس رویکردهای زیادی برای ارائه یک کدگذار پیشگوی کارآمد ارائه شده است که هر یک از این روش ها سعی می کنند تا مقدار نزدیک تری به مقدار واقعی پیکسل پیشگویی نمایند. در این مقاله دیدگاه جدیدی برای بهبود عملکرد کدگذار پیشگو ارائه شده است. در روش پیشنهادی چیدمان تصویر در مرحله پیش پردازش طوری تغییر می یابد تا عملکرد کدگذار پیشگو و به طور کل عملکرد روش های فشرده سازی بدون اتلاف مبتنی بر کدگذار پیشگو بهبود یابند. در روش پیشنهادی ابتدا تصویر به بلوک هایی تقسیم می شود، سپس به کمک الگوریتم ژنتیک یک تبدیل هندسی مناسب بر روی هر بلوک از تصویر اعمال می گردد تا همبستگی بین پیکسل های مجاور افزایش یابد. از اینرو نرخ فشرده سازی JPEG بی اتلاف و CALIC بهبود می یابد. در ادامه تأثیر پیش پردازش تبدیلات هندسی تصویر بر عملکرد کدگذار پیشگو و نحوه انتخاب یک تبدیل هندسی مناسب بر روی بلوک های تصویر به کمک الگوریتم ژنتیک شرح داده می شود.

#### ۴-۱ تأثیر اعمال تبدیلات هندسی به تصویر در عملکرد کدگذار پیشگو

در بسیاری از تصاویر، تبدیلات هندسی سبب می شوند تا چیدمان تصویر طوری تغییر یابد که همبستگی پیکسل های همسایه افزایش یابد. در واقع این کار سبب می شود که مقادیر پیکسل های همسایه هر پیکسل که در پیشگویی مقدار آن پیکسل دخالت دارند، به مقدار واقعی پیکسل نزدیک تر شوند. از اینرو مقدار دقیق تری برای هر پیکسل از تصویر پیشگویی می شود. در نتیجه اعداد کوچک و معدودی برای مقادیر خطا به دست می آیند. از اینرو قابلیت کدگذار

CALIC یکی دیگر از روش های فشرده سازی بدون اتلاف است که از کدگذار پیشگو استفاده می نماید. کدگذار پیشگو در CALIC یک کدگذار نسبتاً پیچیده می باشد که سعی می کند مقداری نزدیک به مقدار واقعی هر پیکسل پیشگویی نماید. در این کدگذار، ابتدا گرادیان عمودی ( $g_v$ ) و گرادیان افقی ( $g_h$ ) تصویر [۲۳] مطابق روابط (۲) و (۳) محاسبه می شوند:

$$g_h = |W - WW| + |N - NW| + |NE - N| \quad (2)$$

$$g_v = |W - NW| + |N - NN| + |NE - NNE| \quad (3)$$

سپس هموار بودن، لبه افقی و لبه عمودی بودن هر پیکسل تصویر با توجه به مقادیر گرادیان بررسی می شوند. بر این اساس مطابق الگوریتم زیر، مقداری برای هر پیکسل از تصویر پیشگویی می شود. سرانجام بر اساس مقدار پیشگویی شده برای هر پیکسل، مطابق رابطه (۱) مقدار خطایی برای هر پیکسل به دست می آید.

<p>If (<math>g_v - g_h &gt; 80</math>) // لبه های افقی تیز</p> <p><math>\hat{P} = W</math></p> <p>else if (<math>g_v - g_h &lt; -80</math>) // لبه های عمودی تیز</p> <p><math>\hat{P} = N</math></p> <p>else {</p> <p><math>\hat{P} = (W + N)/2 + (NE - NW)/4</math></p> <p>// میزان همواری تخمین زده می شود</p> <p>If (<math>g_v - g_h &gt; 32</math>) // لبه های افقی</p> <p><math>\hat{P} = (\hat{P} + W)/2</math></p> <p>else if <math>g_v - g_h &gt; 8</math> // لبه های افقی ضعیف</p> <p><math>\hat{P} = (3\hat{P} + W)/4</math></p> <p>else if <math>g_v - g_h &lt; -32</math> // لبه های عمودی</p> <p><math>\hat{P} = (\hat{P} + N)/2</math></p> <p>else if <math>g_v - g_h &lt; -8</math> // لبه های عمودی ضعیف</p> <p><math>\hat{P} = (3\hat{P} + N)/4</math></p> <p>}</p>
--

#### ۳-۲ کدگذار آنتروپی

آنتروپی بیانگر عدم قطعیت در منبع اطلاعاتی است. افزایش آنتروپی به معنای افزایش عدم قطعیت است، از اینرو اطلاعات بیشتری باید به منبع اختصاص داد. در واقع آنتروپی ابزار مناسبی برای ارزیابی محتوای یک منبع اطلاعاتی است. بر اساس نظریه اطلاعات، متوسط اطلاعات خروجی یک تصویر که بیانگر میزان آنتروپی آن است به صورت رابطه (۴) تعریف می شود [۱]:

$$En = - \sum_{i=1}^n p(a_i) \log_2 p(a_i) \quad (4)$$

در این رابطه  $n$  تعداد سطوح خاکستری موجود در تصویر،  $a_i$  سطح خاکستری  $i$ ام و  $p(a_i)$  احتمال آن سطح خاکستری می باشند. اگر فراوانی سطوح خاکستری موجود در تصویر توزیع یکنواختی نداشته باشد، در این حالت برای رسیدن به نرخ فشرده سازی بالا می توان به نمادهایی که احتمال وقوع آنها بیشتر

شده است. از آنجایی که هر یک از این دو روش فشرده‌سازی، از یک روش کدگذار پیشگوی متفاوت استفاده می‌کنند، پیش‌پردازش بهینه برای هر یک از این دو روش نیز اغلب متفاوت است. در این مثال اگر ابتدا تصویر مرجع را ۹۰ درجه چرخش دهیم و این تصویر پیش‌پردازش شده را به JPEG بی‌اتلاف اعمال نماییم، عملکرد JPEG بی‌اتلاف بهبود می‌یابد. همچنین اگر ابتدا، تصویر مرجع نسبت به محور افقی انعکاس یابد و این تصویر پیش‌پردازش شده به CALIC اعمال گردد، عملکرد CALIC بهبود می‌یابد. در بسیاری از موارد بهتر است این کار به صورت محلی بر روی تصویر انجام گیرد. یعنی ابتدا تصویر به چند بلوک تقسیم شود، سپس یکی از هشت تبدیل هندسی مناسب بر روی هر بلوک به صورت مجزا اعمال گردد تا در نهایت تصویر پیش‌پردازش شده مناسبی برای کدگذار پیشگو به دست آید. در این مقاله از الگوریتم ژنتیک برای انتخاب تبدیل هندسی مناسب هر بلوک از تصویر استفاده شده است.

#### ۲-۴ استفاده از ژنتیک برای تغییر چیدمان تصویر

همان‌طور که در قسمت قبل اشاره شد، با اعمال پیش‌پردازش مناسب بر روی تصویر می‌توان ساختار پیکسل‌های تصویر را طوری تغییر داد تا عملکرد کدگذار پیشگو در پیشگویی مقادیر پیکسل‌های تصویر بهبود یابد. در این مقاله برای تصاویر در اندازه  $512 \times 512$  اندازه بلوک‌ها به صورت تجربی  $64 \times 64$  در نظر گرفته شده است. البته بهتر است که اندازه بلوک‌های تصویر بسته به محتوای هر تصویر به صورت وفقی تعیین گردد که این پیشنهادی برای کار آینده است.

پیشگو و به طور کل قابلیت روش‌های فشرده‌سازی بی‌اتلاف مبتنی بر پیشگو مانند JPEG بی‌اتلاف و CALIC در فشرده‌سازی آن تصویر افزایش می‌یابد. قابل ذکر است تبدیلات هندسی اعمال شده به تصویر باید قابلیت بازگشت‌پذیری داشته باشند. یعنی پس از کدگذاری در مرحله پس‌پردازش با تبدیلی مناسب بتوان تصویر اولیه را بازیابی نمود. از اینرو در این مقاله از تبدیلات هندسی چرخش و انعکاس استفاده شده است. میزان چرخش باید به گونه‌ای باشد که منجر به بزرگتر شدن اندازه تصویر نشود. زیرا افزایش اندازه تصویر منجر به افزایش تعداد پیکسل‌های تصویر می‌شود. در واقع نیاز است این پیکسل‌های اضافه نیز کد گردند.

در روش پیشنهادی از چرخش در جهت‌های ۹۰، ۱۸۰ و ۲۷۰ درجه و انعکاس در جهت محور افقی (x)، محور عمودی (y)، محور  $y=x$  و محور  $y=-x$  استفاده شده است. شکل ۵ نمونه‌ای از یک تصویر مرجع، نسخه فشرده شده آن و نسخه‌های فشرده شده از تصاویر حاصل از هفت تبدیل هندسی مذکور به دو روش JPEG بی‌اتلاف و CALIC را نشان می‌دهد. ممکن است برای برخی از تصاویر، با اعمال این پیش‌پردازش تبدیلات هندسی عملکرد کدگذار پیشگو تضعیف گردد. از اینرو برای برخی از تصاویر نیاز به این تبدیلات هندسی نمی‌باشد. در ادامه برای راحتی، تصویر پیش‌پردازش نشده به عنوان تصویر حاصل از تبدیل هندسی با چرخش صفر درجه در نظر گرفته می‌شود.

همان‌طور که در شکل ۵ مشخص است، هر یک از تبدیلات هندسی مذکور نرخ فشرده‌سازی JPEG بی‌اتلاف و CALIC را تغییر داده‌اند. با مشاهده حجم هشت تصویر فشرده شده در این شکل مشخص است که تبدیل هندسی چرخش ۹۰ درجه، منجر به کمترین میزان حجم برای JPEG بی‌اتلاف و پیش‌پردازش انعکاس نسبت به محور افقی، منجر به کمترین میزان حجم برای CALIC



(ث) چرخش ۲۷۰ درجه، حجم  
۱۷۲،۹۱۳ JPEG  
حجم CALIC ۱۵۴/۹۰۱



(ت) چرخش ۱۸۰ درجه، حجم  
۱۸۲،۳۰۸ JPEG  
حجم CALIC ۱۵۴/۳۰۶



(پ) چرخش ۹۰ درجه، حجم  
۱۷۲،۸۶۳ JPEG  
حجم CALIC ۱۵۴/۹۲۸



(ب) تصویر مرجع فشرده شده،  
حجم JPEG ۱۸۲،۲۳۲  
حجم CALIC ۱۵۴/۱۵۴



(الف) تصویر مرجع، حجم  
۲۵۷



(خ) انعکاس نسبت به محور  $y=-x$ ،  
حجم JPEG ۱۷۲،۸۹۶  
حجم CALIC ۱۵۴/۳۳۲



(ح) انعکاس نسبت به محور  $y=x$ ، حجم  
حجم JPEG ۱۷۲،۸۸۸  
حجم CALIC ۱۵۴/۳۲۲



(ج) انعکاس نسبت به محور عمود، حجم  
حجم JPEG ۱۸۲/۲۷۵  
حجم CALIC ۱۵۳/۶۴۸



(ج) انعکاس نسبت به محور افق،  
حجم JPEG ۱۸۲/۲۳۹  
حجم CALIC ۱۵۳/۶۲۷

شکل ۵ تأثیر پیش‌پردازش به کمک تبدیلات هندسی بر عملکرد JPEG بی‌اتلاف و CALIC (مقادیر حجم بر حسب کیلوبایت هستند).

اینرو اندازه هر کروموزوم  $1 \times 64$  می باشد. تبدیل هندسی مربوط به هر یک از این بلوک های تصویر در ژن های کروموزوم درج شده اند (شکل ۶ الف)). از اینرو به هر بلوک، تبدیل هندسی مربوطه اعمال می گردد. تصویر پیش پردازش شده حاصل در شکل ۶ پ) نشان داده شده است. در روش پیشنهادی در تکرارهای الگوریتم ژنتیک چیدمان تصویر بارها تغییر می کند تا تصویر پیش پردازش شده مناسبی برای JPEG بی اتلاف و یا CALIC به دست آید. پس از انتخاب تصویر پیش پردازش شده مناسب، نیاز است کروموزوم برنده نیز ذخیره گردد تا در مرحله کدگذاری بتوان با پس پردازش مناسب هر بلوک، تصویر اولیه را بازیابی نمود.

جدول ۲ تبدیل هندسی مرتبط با مقدار هر ژن .

مقدار	تبدیل هندسی
۱	بدون پیش پردازش، چرخش صفر درجه
۲	انعکاس نسبت به محور عمود
۳	انعکاس نسبت به محور افقی
۴	چرخش ۱۸۰ درجه
۵	انعکاس نسبت به محور $y=x$
۶	چرخش ۹۰ درجه
۷	چرخش ۲۷۰ درجه
۸	انعکاس نسبت به محور $y=-x$

جدول ۳ پارامترهای الگوریتم ژنتیک.

مقدار	پارامتر
۱۰۰	Population Size
۳۰	Maximum Generation
۰/۵	Mutation Probability

## ۵ نتایج تجربی

عملکرد یک الگوریتم فشرده سازی بی اتلاف براساس میزان حجم تصویر فشرده شده سنجیده می شود. عمق بیتی تصویر (bpp) یک معیار مهم در ارزیابی میزان فشرده سازی تصویر می باشد. عمق بیتی تصویر بیانگر تعداد بیت های تخصیص یافته برای هر پیکسل تصویر است. این معیار برای تصویر خاکستری در محدوده  $[0, 255]$  می باشد. هر چه مقدار این معیار کمتر باشد، یعنی تصویر بیشتر فشرده شده است. عمق بیتی تصویر را می توان مطابق رابطه (۵) محاسبه نمود:

$$bpp = \frac{H}{M \times N} \quad (5)$$

در این رابطه،  $H$  حجم تصویر فشرده شده،  $M$  و  $N$  اندازه سطر و ستون تصویر می باشند. جدول ۴ تأثیر پیش پردازش مذکور را بر روی عملکرد JPEG بی اتلاف و CALIC بر روی دوازده تصویر استاندارد در اندازه  $512 \times 512$  نشان می دهد. همان طور که از نتایج مشخص است، مقدار bpp حاصل از روش پیشنهادی به مراتب کمتر از مقدار bpp روش JPEG بی اتلاف است. همچنین پیش پردازش پیشنهادی توانست عملکرد CALIC را نیز بر روی اغلب تصاویر تا حدودی بهبود بخشد. همان طور که اشاره شد هر دو روش JPEG بی اتلاف و CALIC از کدگذار پیشگو استفاده

در واقع تصاویر در اندازه  $512 \times 512$  به ۱۶ بلوک در اندازه  $64 \times 64$  تقسیم می شود و برای هر بلوک، یک تبدیل هندسی مناسب انتخاب می گردد. تبدیل هندسی برای هر بلوک مناسب است که ضمن افزایش همبستگی پیکسل های همسایه درون آن بلوک، همبستگی پیکسل ها در مرز بین بلوک های همسایه را نیز افزایش دهد. از اینرو قابلیت کدگذار پیشگو در پیشگویی مقادیر پیکسل ها افزایش می یابد. با توجه به مطالب ذکر شده، از آنجایی که تبدیل هندسی مناسب برای هر بلوک، علاوه بر محتوای آن بلوک، به محتوای بلوک های همسایه نیز بستگی دارد، تعداد کل جستجو برای ۱۶ بلوک تصویر در اندازه  $512 \times 512$  برابر  $8^{16}$  حالت می باشد. در واقع اگر از جستجوی کامل برای تعیین تبدیل هندسی مناسب برای بلوک های تصویر استفاده شود، پیدا کردن تبدیل های مناسب برای بلوک های یک تصویر در اندازه  $512 \times 512$  با کامپیوتر پنتیوم با پردازنده ۲/۶ مگاهرتز ۲ هسته ای و RAM چهار گیگا بایت با نرم افزار Matlab بیشتر از ۱۵ ساعت طول می کشد که عملاً غیر قابل استفاده است. از اینرو برای کاهش زمان محاسبات از الگوریتم ژنتیک استفاده می شود، تا ضمن کاهش زمان کدگذاری با انجام تبدیلات هندسی به کاهش مناسب در فشرده سازی دست یابد.

برای هر بلوک از تصویر یک ژن در کروموزوم لحاظ شده است که مقادیر آن در محدوده صحیح یک تا هشت است. از اینرو برای هر بلوک از تصویر، یکی از هشت تبدیل هندسی انتخاب می شود تا با توجه به آن، پیش پردازشی بر روی هر بلوک اعمال گردد. جدول ۲ تبدیل هندسی مربوط به هر یک از مقادیر یک تا هشت را نشان می دهد. با توجه به مقادیر هر کروموزوم، یک پیش پردازش بر روی تصویر انجام می شود. همان طور که اشاره شد، از آنجایی که کدگذارهای پیشگو در دو روش JPEG بی اتلاف و CALIC متفاوت هستند، نیاز است پیش پردازش مناسبی برای هر یک از این دو روش انتخاب گردد. اگر هدف، بهبود عملکرد JPEG بی اتلاف باشد، در تابع برازندگی از JPEG بی اتلاف استفاده می شود. اگر هدف، بهبود عملکرد CALIC باشد، در تابع برازندگی از این روش استفاده می شود. در واقع در تابع برازندگی هر یک از تصاویر پیش پردازش شده با یکی از این دو روش فشرده سازی، فشرده می گردد و حجم تصویر پیش پردازش شده به عنوان مقدار تابع برازندگی الگوریتم ژنتیک در نظر گرفته می شود. هر چه حجم تصویر فشرده شده کمتر باشد، کروموزوم مربوطه، کروموزوم برتر می باشد. یعنی با پیش پردازش تصویر مطابق با مقادیر کروموزوم برنده، عملکرد روش فشرده سازی بی اتلاف بهبود می یابد. در این مقاله پارامترهای الگوریتم ژنتیک به صورت مقادیر درج شده در جدول ۳ در نظر گرفته شده اند.

مقادیر یک کروموزوم نمونه و تصویر پیش پردازش شده براساس مقادیر آن در شکل ۶ نشان داده شده است. در این مثال، تصویر ورودی در اندازه  $512 \times 512$  و اندازه بلوک ها  $64 \times 64$  می باشد. در واقع، تصویر ورودی به ۱۶ بلوک تقسیم شده است، از

می‌کنند، ولی کدگذار پیشگو در CALIC یک کدگذار پیچیده‌ای است که ابتدا با محاسبه گرادیان تصویر، نوع هر پیکسل (لبه، یا میزان همواری) را مشخص می‌کند. سپس با توجه به نوع پیکسل، مقدار دقیق‌تری را برای هر پیکسل پیشگویی می‌نماید. درحالی‌که کدگذار پیشگو در JPEG بی‌اتلاف یک کدگذار ساده است که بدون توجه به نوع پیکسل، پیشگویی یکسانی برای تمامی پیکسل‌ها انجام می‌دهد. از اینرو پیش‌پردازش پیشنهادی تا حدود زیادی نرخ فشرده‌سازی JPEG بی‌اتلاف را بهبود می‌دهد. همچنین با وجود کدگذار پیشگوی پیچیده در CALIC، پیش‌پردازش پیشنهادی توانست روش CALIC را هم تا حدودی بهبود بخشد. همان‌طور که اشاره شد، وقتی بر روی بلوکی تبدیل هندسی اعمال می‌شود، اغلب پیکسل‌های همسایه هر پیکسل در آن بلوک تغییر می‌کند. برای نمونه اگر لبه عمودی در بلوک باشد، با چرخش ۹۰ درجه، لبه

افقی در آن بلوک به وجود می‌آید. از اینرو چیدمان جدید پیکسل‌ها، بر روی مقدار پیشگویی شده هر پیکسل تأثیر دارند. در نتیجه با توجه به محتوای هر تصویر، تأثیر پیش‌پردازش پیشنهادی بر روی هر تصویر متفاوت است. به‌طور کلی، هدف این مقاله این است که به‌جای ارائه یک روش پیشگوی جدید، از پیش‌پردازش برای بهبود روش‌های فشرده‌سازی استفاده نماید.

تأثیر پیش‌پردازش پیشنهادی بر روی روش فشرده‌سازی مطرح شده در مرجع [۱۸] بررسی شده است که چهار نمونه از آن در جدول ۵ نشان داده شده است. در واقع هدف، بهبود عملکرد روش ارائه شده در مرجع [۱۸] می‌باشد، از اینرو در تابع برازندگی از این روش فشرده‌سازی استفاده می‌شود.

۶	۸	۶	۶	۶	۶	۷	۷	۶	۵	۴	۵	۱	۱	۴	۳
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

الف) نمایش مقادیر یک کروموزوم



ب) تصویر پیش‌پردازش شده



ب) تصویر مرجع

شکل ۶ نمونه‌ای از عملکرد یک کروموزوم برای تغییر چیدمان تصویر به روش پیشنهادی.

جدول ۴ تأثیر پیش‌پردازش پیشنهادی بر روی عملکرد JPEG بی‌اتلاف و CALIC بر اساس معیار .bpp

	پیش‌پردازش پیشنهادی + JPEG بی‌اتلاف	JPEG بی‌اتلاف	میزان بهبود JPEG بی‌اتلاف	پیش‌پردازش پیشنهادی CALIC +	CALIC	میزان بهبود CALIC
Lena	۴,۸۰۹۲	۵,۱۶۰۷	۰,۳۵۱۵	۴,۱۱۷۲	۴,۱۱۹۴	۰,۰۰۲۲
Zelda	۴,۴۵۳۴	۴,۸۷۸۸	۰,۴۲۵۴	۳,۸۷۶۹	۳,۸۷۶۹	۰,۰
Couple	۵,۲۶۹۰	۵,۵۷۴۲	۰,۳۰۵۳	۴,۶۲۴۵	۴,۶۲۴۵	۰,۰
Mandrill	۶,۴۴۶۱	۶,۴۴۶۱	۰,۰۲۵۹	۵,۸۷۲۹	۵,۸۹۷۶	۰,۰۲۴۷
Boats	۵,۳۵۲۵	۵,۶۹۴۸	۰,۳۴۲۲	۴,۰۰۷۹	۴,۰۱۶۳	۰,۰۰۸۳
Clown	۴,۷۹۹۲	۵,۵۳۱۵	۰,۷۳۲۳	۴,۰۰۳۶	۴,۰۰۳۶	۰,۰
Woman with blond hair	۴,۶۰۰۴	۴,۹۰۶۹	۰,۳۰۶۵	۳,۹۹۳۲	۳,۹۹۷۴	۰,۰۰۴۲
Woman with dark hair	۳,۵۴۷۱	۳,۸۲۲۵	۰,۲۷۵۴	۱,۶۸۲۷	۱,۶۸۲۷	۰,۰
Barbara	۵,۵۰۹۳	۶,۱۸۹۱	۰,۶۷۹۸	۴,۵۲۴۸	۴,۵۳۱۷	۰,۰۰۶۹
Einstein	۴,۶۱۱۷	۴,۷۸۳۲	۰,۱۷۱۵	۳,۸۴۵۹	۳,۸۴۵۹	۰,۰
Goldhill	۵,۲۴۹۱	۵,۳۹۳۷	۰,۱۴۴۶	۴,۶۳۰۱	۴,۶۴۴۳	۰,۰۱۴۱
Elaine	۵,۴۳۲۳	۵,۴۳۲۳	۰,۰	۴,۸۱۳۰	۴,۸۱۴۳	۰,۰۰۱۲



کدگذار JPEG بی اتلاف و CALIC می باشد. ولی مجموع زمان محاسبات پس پردازش و کدگشا در روش پیشنهادی به زمان کدگشا در JPEG بی اتلاف و CALIC نزدیک است. قابل ذکر است، در کاربردهایی مانند سیستم های بایگانی داده، که فرآیند فشرده سازی به عنوان یک کار پس زمینه محسوب می شود، اتلاف زمانی فرآیند کدگذار قابل تحمل است. ولی سریع بودن زمان کدگشا از اهمیت بالایی برخوردار است. همان طور که اشاره شد، اگر به جای الگوریتم ژنتیک، از جستجوی کامل برای انتخاب تبدیل هندسی مناسب برای بلوک های تصویر استفاده شود، این جستجو ساعت ها طول می کشد. همچنین اگر چه الگوریتم ژنتیک توانست زمان کدگذاری را نسبت به کدگذاری در جستجوی کامل بسیار کاهش دهد، ولی می توان روشی برای بهبود این زمان کدگذاری نیز پیشنهاد داد.

## ۶ نتیجه گیری

در این مقاله یک روش پیش پردازش برای بهبود عملکرد کدگذار پیشگو ارائه شده است. در این روش ابتدا تصویر به بلوک هایی تقسیم می شود. سپس به کمک الگوریتم ژنتیک، یک تبدیل هندسی مناسب برای هر بلوک از تصویر انتخاب می گردد تا همبستگی پیکسل های همسایه افزایش یابد. بدین طریق کدگذار پیشگو قادر به پیشگویی دقیق تری برای پیکسل ها می باشد و عملکرد این کدگذار و به طور کلی عملکرد روش های فشرده سازی بدون اتلاف مبتنی بر کدگذار پیشگو همچون JPEG بی اتلاف و CALIC بهبود می یابد. در روش پیشنهادی، اندازه بلوک های تصویر ثابت در نظر گرفته شد، برای بهبود روش پیشنهادی می توان اندازه بلوک های تصویر را بسته به محتوای هر تصویر به صورت وفقی تعیین نمود.

همان طور که اشاره شد روش مطرح شده در [۱۸] یک روش فشرده سازی بدون اتلاف مبتنی بر کدگذار پیشگو است. در این روش، کدگذار پیشگویی سلسله مراتبی پیشنهاد شده است تا بتواند پیشگویی دقیق تری برای پیکسل ها انجام دهد. همان طور که از نتایج مشخص است، مقدار bpp روش پیشنهادی در مواردی برابر یا کمتر از مقدار bpp روش [۱۸] می باشد. قابل ذکر است نتایج مطرح شده در جدول ۵ برای تصاویر رنگی می باشد که از مدل رنگ YCbCr استفاده شده است. مدل رنگ YCbCr مدل رنگ مناسب برای فشرده سازی تصویر است که در اغلب استانداردهای فشرده سازی از آن استفاده می شود [۲۴].

جدول ۵ تأثیر پیش پردازش پیشنهادی بر روش [۱۸] بر اساس معیار  $\text{bpp}$ .

	پیش پردازش پیشنهادی + مرجع [۱۸]	مرجع [۱۸]	میزان بهبود مرجع [۱۸]
Lena	۱۳,۶۴۶۱	۱۳,۶۴۶۱	۰,۰
Peppers	۱۵,۲۱۰۲	۱۵,۲۱۰۲	۰,۰
Mandrill	۱۸,۴۸۵۰	۱۸,۵۲۳۰	۰,۰۳۸۰
Barbara	۱۱,۲۲۲۱	۱۱,۸۴۳۱	۰,۶۲۰۹

جدول ۶ روش پیشنهادی را از نظر میانگین زمانی با روش JPEG بی اتلاف و CALIC بر روی چهار تصویر استاندارد در سه اندازه  $۲۵۶ \times ۲۵۶$ ،  $۵۱۲ \times ۵۱۲$  و  $۱۰۲۴ \times ۱۰۲۴$  مقایسه می نماید. تمامی این روش ها به کمک کامپیوتر پنتیوم با پردازنده ۲/۶ مگاهرتز ۲ هسته ای و RAM چهار گیگا بایت با نرم افزار ۲۰۱۴ Matlab انجام شده اند. قابل ذکر است، کلیه زمان های درج شده در جدول ۶ با دو دستور tic و toc برنامه Matlab به دست آمده اند.

همان طور که مشخص است مجموع زمان محاسبات پیش پردازش و کدگذار روش پیشنهادی به مراتب بیشتر از زمان

جدول ۶ مقایسه روش پیشنهادی با JPEG بی اتلاف از نظر میانگین زمان محاسبات بر حسب ثانیه بر روی چند تصویر استاندارد.

اندازه تصویر	JPEG بی اتلاف		روش پیشنهادی + JPEG بی اتلاف		CALIC		روش پیشنهادی + CALIC	
	کدگشا	کدگذار	کدگشا و پس پردازش	کدگذار و پیش پردازش	کدگشا	کدگذار	کدگشا و پس پردازش	کدگذار و پیش پردازش
$۲۵۶ \times ۲۵۶$	۰,۰۰۴	۰,۰۳۰	۰,۰۱	۵۶,۷۵	۰,۱۰۳	۰,۱۳۲	۰,۱۷۵	۳۳,۰۹۱
$۵۱۲ \times ۵۱۲$	۰,۰۹	۰,۰۴۸	۰,۰۲۹	۱۲۹,۵	۰,۲۷۵	۰,۲۹۳	۰,۳۳۳	۸۲۲,۸۰
$۱۰۲۴ \times ۱۰۲۴$	۰,۰۲۶	۰,۱۱۷	۰,۱۰۷	۴۰۹,۷۸	۰,۷۳۵	۰,۸۳۲	۰,۹۲۰	۲۵۵۲,۷

- International conference Visualization, Imaging, And Image Processing, Spain, 2004.
- [15] Nasr Esfahani, E., Samavi, S., Karimi, N., Shirani, S., "Near-Lossless Image Compression Based on Maximization of Run Length Sequences", International Conference on Information Processing, pp. 177-180, 2007.
- [16] Nasr Esfahani, E., Samavi, S., Karimi, N., "Near lossless Image Compression by Local Packing of Histogram", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1197-1200, 2008.
- [17] Yerva, S., Nair, S., Kutty, K., "Lossless Image Compression based on Data Folding" IEEE International Conference on Recent Trends in Information Technology, pp. 999-1004, 2011.
- [18] Seyun Kim, S., Cho, N. I., "Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression", IEEE Transactions on Image Processing, Vol. 23, No. 1, pp. 445-449, 2014. Available: <http://ispl.snu.ac.kr/light4u/project/LCIC>, accessed: Octobr 2015.
- [19] OwenZhao, X., HenryHe, Z., "Lossless Image Compression Using Super-Spatial Structure Prediction", IEEE Signal Processing Letters, Vol. 17, No. 4, pp. 383-386, 2010.
- [20] Yang, L., He, X., Zhang, G., Qing, L., Che, T., "A low complexity block-based adaptive lossless image compression", Optik, Vol. 124, pp. 6545-6552, 2013.
- [21] Wu, J., Kong, W., Mielikainen, J., Huang, B., "Lossless Compression of Hyperspectral Imagery via Clustered Differential Pulse Code modulation with Removal of Local Spectral Outliers", IEEE Signal Processing Letters, Vol. 22, No. 12, pp. 2194-2198, 2015.
- [22] Syahrul, E., "Lossless and Nearly-Lossless Image Compression Based on Combinatorial Transforms", Ph.D. Thesis, University of Burgundy, 2011.
- [23] Wu, X., "An algorithmic study on lossless image compression", Proceedings of the Conference on Data Compression, pp. 150-159, 1996.
- [24] Wang, Ch., Xie, S., Zhou, X., "Bayer Patterned Image Compression Based on Structure Conversion and APBT", International Journal of Multimedia and Ubiquitous Engineering Vol. 10, No. 2, pp. 333-340, 2015.
- [1] Gonzalez, R. C., Woods, R. E., "Digital Image Processing", 2nd Edition, Pearson Prentice Hall, ISBN: 81-7758-168-6, 2005.
- [2] Cintra, R. J., Bayer, F. M., Tablada, C. J. "Low-Complexity 8-Point DCT Approximations Based on Integer Functions", Signal Processing, Vol. 99, pp. 201-214, 2014
- [3] Kim, J., Kim, J., Kyung, Ch. M., "A Lossless Embedded Compression Algorithm for High Definition Video Coding", IEEE International Conference on Multimedia and Expo, pp. 193-196, 2009.
- [4] Ernawan, F., Abu, N., A., Suryana, N., "Integrating a Smooth Psychovisual Threshold into an Adaptive JPEG Image Compression", Journal of computers, Vol. 9, No. 3, pp. 644-653, 2014.
- [5] Sanchez, V., Bartrina-Rapesta, J., "Lossless Compression of Medical Images based on Hevc intra Coding", IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pp. 6672-6676, 2014.
- [6] Al-laham, M., Emary, M., "Comparative Study between Various Algorithms of Data Compression Techniques", International Journal of Computer Science and Network Security (IJCSNS), Vol. 7, No. 4, pp. 281-291, 2007.
- [7] Clunie, D. A., "Lossless compression of grayscale medical images: effectiveness of traditional and state-of-the-art approaches", Medical Imaging, Engineering and Clinical Issues, Vol. 74, 2000.
- [8] Weinberger, M. J., Seroussi, G., Sapiro, G., "LOCO-I: a low complexity, context-based, lossless image compression algorithm", Data Compression Conference, pp. 140-149, 1996.
- [9] Skodras, A., Christopoulos, C., Ebrahimi, T., "The JPEG2000 still image compression standard", IEEE Signal Processing Magazine, Vol. 18, pp. 36-58, 2001.
- [10] Wu, X., Memon, N., "Context-based, Adaptive, Lossless Image Coding", IEEE Transactions on Communications, Vol. 45, No. 4, pp. 437-444, 1997. Available: <https://github.com/gameclosure/gcif/tree/master/refs/calic>: June 2014.
- [11] Lastri, C., Aiazzi, B., Alparone, L., Baronti, S., "Virtually Lossless Compression of Astrophysical Images", Journal on Applied Signal Processing, Vol. 15, pp. 2521-2535, 2005.
- [12] Kim, S. J., Hwang, D. Y., Yoo, G. H., "A Preprocessing Algorithm for Efficient Lossless Compression of Gray Scale Images", IEEE International Conference on Control, Automation and Systems, 2005.
- [13] Pinho, A. J., Neves, A., "Block Based Histogram Packing of Color-Quantized Images", IEEE International Conference on Multimedia & Expo, pp. 341-344, 2003.
- [14] Pinho, A. J., Neves, A., "Variable Size Block-Based Histogram Packing for Lossless Coding of Color-Quantized Images", Proceedings of the Fourth



سکینه اسدی امیری دانشجوی دکتری مهندسی کامپیوتر گرایش هوش مصنوعی از دانشگاه شاهرود است. ایشان مدرک کارشناسی ارشد خود را از دانشگاه شاهرود در رشته مهندسی کامپیوتر گرایش هوش مصنوعی در سال ۱۳۸۹ و مقطع کارشناسی خود را از دانشگاه پیام نور بابل در رشته مهندسی کامپیوتر گرایش نرم افزار در سال ۱۳۸۶ اخذ نموده است. زمینه تحقیقاتی ایشان پردازش تصویر و شبکه عصبی می باشد.



حمید حسن پور در سال ۱۳۸۳ تحصیلات مقطع دکتری خود را در مهندسی کامپیوتر گرایش پردازش سیگنال در دانشگاه صنعتی کوئینزلند استرالیا به اتمام رساند. ایشان مدرک کارشناسی ارشد خود را در سال ۱۳۷۵ در گرایش هوش مصنوعی از دانشگاه صنعتی امیرکبیر اخذ نمود و مدرک کارشناسی خود را در سال ۱۳۷۲ در مهندسی کامپیوتر گرایش سخت افزار از دانشگاه علم و صنعت ایران دریافت نموده است. ایشان در حال حاضر استاد دانشکده مهندسی کامپیوتر دانشگاه شاهرود هستند و زمینه تحقیقاتی ایشان پردازش سیگنال، پردازش تصویر و داده کاوی است.

Archive