ارائه راهکار جدید برای فشرده‌سازی تصاویر رنگی دیجیتال مبتنی بر تکنیک شبه رنگ آمیزی با استفاده از خوشه‌بندی رنگ‌ها

محمد زارع[۱]، پیروز شمسی نژاد[۲]، افسانه ساعی آرزومند[۳]، حسن حسینیان[۴]

[۱] گروه مهندسی کامپیوتر موسسه آموزش عالی آپادانا شیراز
arioobarzan@hotmail.com

[۲] گروه مهندسی کامپیوتر دانشگاه صنعتی شیراز
p.shamsinejad@sutech.ac.ir

[۳] تیم فنی مهندسی آریوبرزن شیراز
info@afsanesaee.ir

[۴] تیم فنی مهندسی آریوبرزن شیراز
hasan.hosseinian@gmail.com

چکیده

فشرده‌سازی فایل‌ها در کامپیوتر از اهمیت ویژه ای برخوردار است، دلیل این امر کاهش هزینه در ذخیره‌سازی، ارسال در شبکه‌های کامپیوتری و موارد دیگر می باشد. بیشترین حجم استفاده از اینترنت در دنیا مربوط به تصاویر می باشد. در این مقاله ما سعی داریم تا با الگوریتم پیشنهادی، راهکاری جدید برای فشرده‌سازی این دسته از فایل‌ها عنوان کنیم. تکنیک ارائه شده در این مقاله به این صورت می باشد که در ابتدا تصاویر را با تکنیک K-Means و با توجه به رنگ‌های موجود در پیکسل‌های آن که در مُد رنگی RGB تعریف می‌شود با تعداد خوشه‌های ثابت، خوشه‌بندی و ذخیره گردد، سپس یک عکس خاکستری از تصویر اصلی را ساخته و آن را نیز ذخیره می‌شود، تصویر خوشه بندی شده به عنوان راهنمای رنگی و تصویر خاکستری[×†] به عنوان راهنمای مکانی تصویر می‌باشند، پس از ساخت این دو تصویر توسط تکنیک‌های بسته بندی فایل[‡]، این دو فایل به یک فایل تبدیل می شود تا سربار[§] ارسال در شبکه یا ذخیره‌سازی در رسانه‌ها به حداقل کاهش یابد، ذخیره سازی این دو تصویر حجم بسیار کمتری نسبت به تصویر اولیه خواهد داشت. این نوع فشرده‌سازی مانند عموم تکنیک‌های رایج از نوع با اتلاف می باشد. برای بازیابی تصویر اولیه با استفاده از فایل ذخیره شده توسط تکنیک‌های شبه رنگ آمیزی تصاویر سیاه و سفید[××]، تصویر بازسازی می‌گردد. در نهایت بازیابی تصویر اولیه از طریق مُد رنگی HSV انجام می شود و در آخر روش پیشنهاد شده در این مقاله را با روش‌های دیگر مقایسه خواهد شد. نتایج به دست آمده از الگوریتم پیشنهادی بیانگر کاهش محسوس حجم تصاویر با کیفیتی نسبتا مناسب می باشد.

کلمات کلیدی

Compression, Image Processing, Clustering, K-Means, Pseudo-Coloring

---

× Grayscale

‡ File Packaging

§ Overhead

×× Pseudo-Coloring

۱- مقدمه

هر تصویر دیجیتال رنگی را می توان توسط یک ماتریس دو بعدی با اعداد ۸ بیتی و در سه لایه RGB تعریف کرد. کوچکترین واحد ذخیره سازی تصویر پیکسل می باشد که در حالت عادی برای ذخیره سازی هر پیکسل احتیاج به سه عدد ۸ بیتی یعنی ۲۴ بیت دارد، با این تفسیر اگر یک تصویر با ابعاد ۱۰۰۰ پیکسل در ۱۰۰۰ پیکسل وجود داشته باشد ۲۴ میلیون بیت (حدود ۳ مگابایت) برای ذخیره سازی احتایج می باشد. در عصر امروز بیشترین استفاده از اینترنت مربوط به تصاویر می باشد که از این رو فشرده سازی آن ها بسیار حایز اهمیت می باشد. برای فشرده سازی تصاویر دیجیتال رنگی راهکارهای زیادی ارائه شده است که در ۲ حالت کلیِ با اتلاف[††] و بدون اتلاف[‡‡] می گنجند. در حالت با اتلاف احتمال از بین رفتن پیکسل ها وجود دارد (e.g. JPEG) اما در حالت بدون اتلاف پیکسل ها تغییری نمی کنند (e.g. PNG) [۱،۲،۳]. فشرده سازی تصاویر نوعی از فشرده سازی داده ای می باشد که روی تصاویر اعمال شده و هدف آن کم کردن حجم تصاویر می باشد، روش های زیادی برای فشرده سازی تصاویر ارائه شده است که از تکنیک های مختلفی استفاده کرده اند، برای بخش بندی این تکنیک ها و با توجه به خروجی به دست آمده تکنیک ها را به دو دسته کلی با اتلاف و بدون اتلاف تقسیم کرده اند، تکنیک هایی که خروجی آن ها بدون اتلاف می باشد خروجی تصویر با عکس ورودی کاملا یکسان بوده و فقط حجم تصویر کاهش یافته است در حالی که در تکنیک های با اتلاف تصاویر ورودی و خروجی کاملا یکسان نبوده اما سعی در از دست دادن کمترین داده می باشد. طبیعی است که تکنیک هایی که با اتلاف هستند از انعطاف بیشتری برخوردار بوده و در کاهش حجم موفق تر هستند، تکنیک JPEG و JPEG2000 از این دسته می باشند. بعد از اعمال این تکنیک ها به نظر می آید تصویر به فشرده ترین حالت ممکن خود رسیده است چرا که با تکنیک های فشرده سازی داده ای هم دیگر نمیتوان حجم آن ها را کاهش داد. در این مقاله راهکار ارائه شده در حالت با اتلاف می باشد. تکنیک های زیادی برای این حالت وجود دارد اما به خوبی قابل درک است که فشرده سازی بیش از یک مقدار معین غیرقابل انجام خواهد بود[۴]. خوشه بندی تصاویر تکنیک خوبی برای قطعه بندی تصاویر از لحاظ رنگی می باشد[۶،۷]، برای فشرده سازی تصاویر ما در این مقاله سعی می کنیم که با استفاده از روش های خوشه بندی، مرز میان رنگ های بیشتر به کار رفته در تصویر را پیدا کنیم تا یک نقشه با تعداد محدودی از رنگ ها را از تصویر مورد نظر را ترسیم کنیم[۸]، برای این کار در ابتدا عکس دریافتی را با الگوریتم K-Means و بر اساس مشخصات رنگی پیکسل ها که در سه بعد R, G و B می باشند خوشه بندی کرده و سپس تصویری از خوشه های رنگی را ذخیره می کنیم [۹،۱۰،۱۱]. تصویر اصلی را به یک عکس در مد خاکستری تبدیل می کنیم و با استفاده از تصویر مد خاکستری به دست آمده به همراه تصویر خوشه بندی رنگی ذخیره شده با الگوریتم های تکنیک‌های شبه رنگ آمیزی تصاویر سیاه و سفید سعی در بازیابی تصویر اصلی می کنیم [۱۲،۱۳،۱۴،۱۵]. الگوریتم های شبه رنگ آمیزی هدف رنگی سازی تصاویر مد خاکستری را دنبال می کنند، چگونگی رنگ کردن هر پیکسل در این الگوریتم ها مورد بحث قرار می گیرد، نکته مهم در این الگوریتم ها این است که چگونه عملیات تبدیل یک عدد ۸ بیتی که در مد خاکستری تعریف شده به ۳ عدد ۸ بیتی در مد رنگی را انجام دهیم، برای این کار تکنیک های زیادی تعریف شده که در ادامه راهکار پیشنهادی خود را ارائه می دهیم، در بخش ۲ روش پیشنهادی به کار رفته در این مقاله بررسی می شود، در بخش ۳ موارد مربوط به پیاده سازی را مورد بحث قرار می دهیم و در انتها در بخش ۴ نتایج به دست آمده از تکنیک های عنوان شده را با روش های دیگر مقایسه می کنیم.

۲- روش پیشنهادی

الگوریتم روش پیشنهادی به کار رفته در این مقاله به صورت زیر می باشد:

۱- گرفتن تصویر رنگی از ورودی
۲- خوشه بندی تصویر و ذخیره سازی تصویر خوشه بندی شده
۳- ساخت نمونه تصویر مد خاکستری از تصویر اولیه
۴- ادغام و بسته بندی فایل خوشه بندی شده و مد خاکستری
۵- بازیابی تصویر اولیه و نمایش

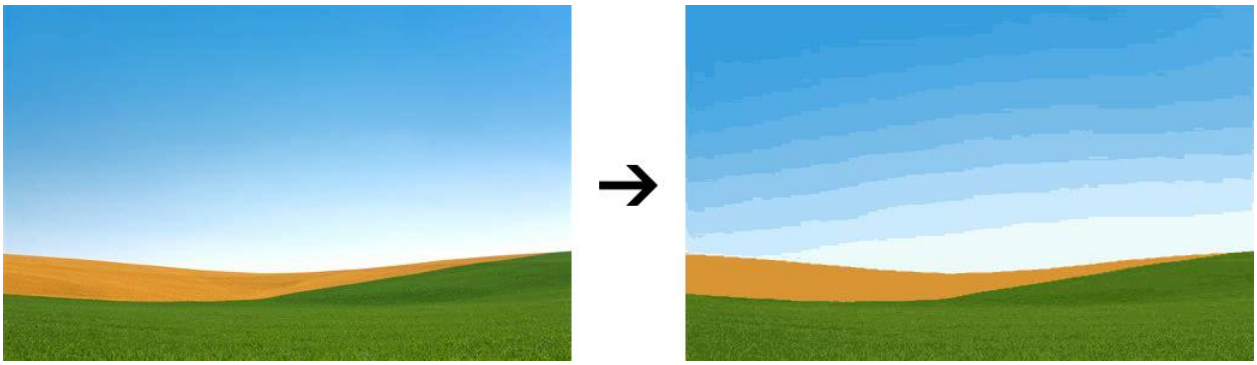
۲-۱- خوشه بندی

خوشه بندی داده[§§] روشی برای پیدا کردن داده های شبیه به هم می باشد که از روی متعلقات داده ها این کار را انجام داده و هدف آن نسبت داده هر داده به یک خوشه می باشد، یکی از تکنیک های خوشه بندی K-Means نام دارد.

---

†† Lossy
‡‡ Lossless

§§ Clustering



شکل ۱: استفاده از تکنیک K-Means و خروجی آن

K-Means -۲-۲

تکنیک K-Means با توجه به تعداد K (خوشه ها) که از ورودی گرفته داده ها را خوشه بندی می کند. داده های تصویر در پیکسل های تصویر می باشند و صفت های آن ها مقادیر ذخیره شده در لایه های RGB یعنی داده ها در تصویر سه بعدی می باشند. بعد از اینکه هر پیکسل با توجه به رنگ آن در یک خوشه قرار گرفت آن خوشه را با توجه به میانگین رنگی آن خوشه رنگ می کنیم، تصویر ۳ خروجی یک تصویر را با K=15 که توسط تکنیک K-Means خوشه بندی شده نشان می دهد.

Gray مد -۳-۲

مد خاکستری یا همان عکس های سیاه و سفید یک مد خاص از تصاویر دیجیتال است که در آن سه لایه RGB به یک لایه خاکستری تبدیل شده و حوزه رنگی تصاویر محدود می شود. برای تبدیل سه لایه RGB به خاکستری از فرمول نمایش داده در شکل ۴ استفاده می کنیم.

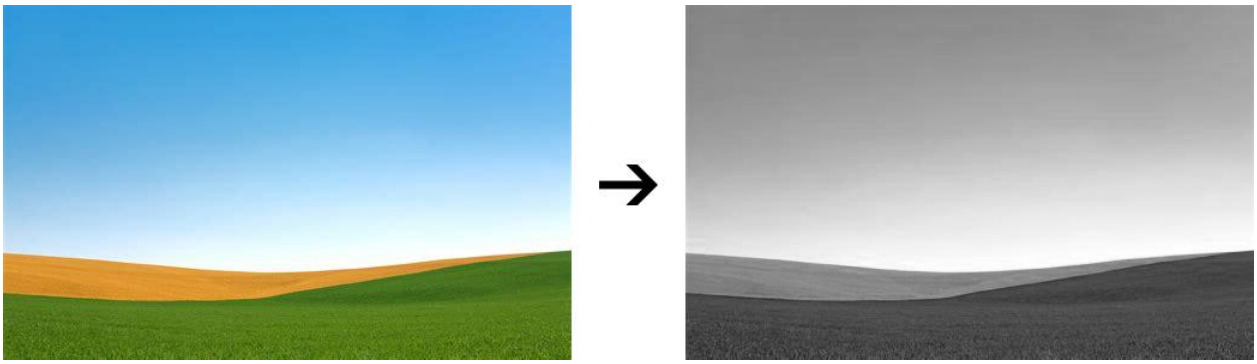
$$0.298 * R + 0.587 * G + 0.114 * B = gray$$

شکل ۲

در فرمول ذکر شده R به معنای عدد ذخیره شده در لایه قرمز، G عدد ذخیره شده در لایه سبز و B عدد ذخیره شده در لایه آبی می باشد.
بعد از اعمال فرمول فوق روی تصویر خروجی به دست آمده در تصویر ۵ نشان داده شده است.

۴-۲- ذخیره سازی

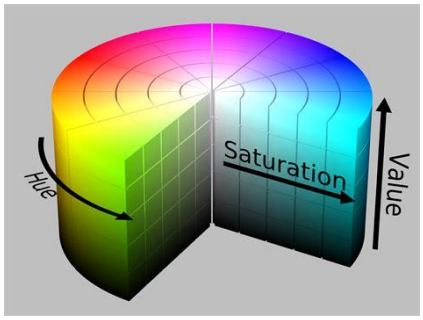
عکس های به دست آمده از خوشه بندی و مد خاکستری را با هم ادغام و در قالب یک فایل، با استفاده از تکنیک بسته بندی فایل ذخیره می کنیم. روش پیشنهادی ما برای این مورد تکنیک ZipPackaging می باشد که با استفاده از الگوریتم زیپ کردن فایل ضمن ادغام کردن قایل ها و نگهداری آن ها در یک فایل از یک الگوریتم فشرده سازی داده نیز استفاده می کند.



شکل **۳:** فرمول تبدیل سه لایه RGB **به** Gray

۲-۵- بازیابی تصویر اولیه

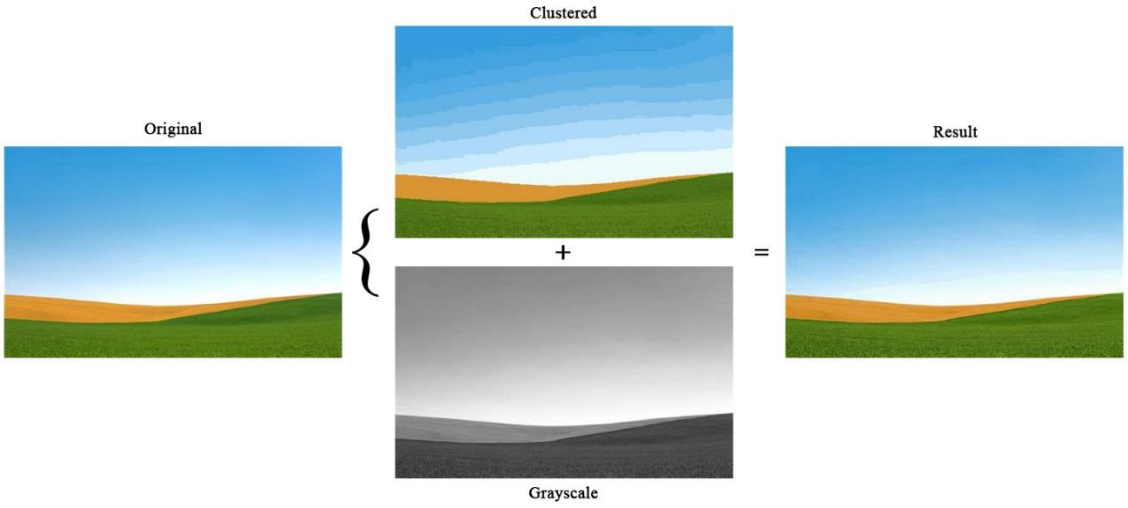
برای بازیابی تصویر اولیه ابتدا تصاویر ذخیره شده را به مد HSV می آوریم. مد HSV یک مد رنگی است که در آن رنگ ها روی یک استوانه در فضای سه بعدی تعریف می شوند، H مخفف کلمه HUE می باشد که با آن زاویه مکان را مشخص می کنیم و حوزه تغییرات آن از ۰ تا ۳۵۹ می باشد، S مخفف Saturation (اشباع رنگ) است و میزان فاصله مکان از مرکز استوانه تا محیط را به درصد بیان می کند که می توانیم از ان به عنوان شدت رنگ نیز نام ببریم، مکان رنگ هرچه به محیط نزدیک تر باشد شدت بیشتری دارد و هر چه به مرکز نزدیک تر باشد رنگ کمتری دارد.V مخفف کلمه Value (ارزش رنگ) به معنای مقدار و شدت نور می باشد، که به درصد بیان می شود این عدد هر چه به صفر نزدیک تر باشد رنگ تیره تر می باشد. تصویر شماره ۶ مد رنگی HSV را نشان می دهد. این تصویر از https://en.wikipedia.org/wiki/HSL_and_HSV برگرفته شده است.



شکل ۴: مُد رنگی HSV

۲-۶- Pseudo-Coloring

اصطلاح تکنیک‌های شبه رنگ آمیزی تصاویر سیاه و سفید Pseudo-Coloring به معنای رنگی سازی تصاویر سیاه و سفید می باشد، یکی از الگوریتم های این تکنیک استفاده از یک رنگ پیشنهادی است، در این مقاله و با توجه به اینکه ما عکس خوشه بندی شده را در اختیار داریم رنگ پیشنهادی را از عکس خوشه بندی شده انتخاب می کنیم و سپس با منطبق کردن تصاویر سیاه و سفید و خوشه بندی شده عکس اولیه را بازیابی می کنیم. برای این کار مقدار Value را از عکس سیاه و سفید و مقادیر Hue و Saturation را از عکس خوشه بندی شده می گیرد و آن ها در تصویر جدیدی ذخیره کرده و نمایش می دهد. روند کلی کار را در تصویر ۷ مشاهده می کنید.



شکل ۵ : روند کلی کار

۳- پیاده سازی و تحلیل نتایج

تکنیک عنوان شده در این مقاله با زبان C# و در محیط ویژوال استودیو ۲۰۱۵ پیاده سازی شده است. سیستم سخت افزاری مورد استفاده دارای یک CPU مدل G3220 دارای ۲ هسته با قدرت پردازش ۳ گیگاهرتز می باشد، میزان رم ۸ گیگا بایت می باشد و سیستم عامل ویندوز ۱۰ بوده است. نتایج حاصل شده بر روی ۱۲ تصویر در جداول و تصاویر انتهای مقاله آورده شده است که با مقایسه بین روش پیشنهادی ما، JPEG، JPEG با کیفیت ۵۰٪ و PNG از لحاظ: حجم فایل فشرده سازی شده، زمان ساخت و زمان بازیابی فایل فشرده شده و همچنین کیفیت تصویر نهایی می باشد. در گزارش های موجود به شکل نمودار میله ای رنگ آبی برای فرمت PNG، رنگ قرمز برای JPEG کیفیت ۱۰۰٪، رنگ زرد برای فرمت JPEG با کیفیت ۵۰٪ و رنگ بنفش نمایانگر روش پیشنهادی ما در این مقاله می باشد.

PSNR -۱-۳

معیار سنجش شباهت در عکس ها تکنیک های گوناگونی دارد که یکی از آن ها PSNR مخفف Peak signal-to-noise ratio است و از فرمول موجود در معادله ۱ به دست می آید. البته در اینجا MAX برابر ۲۵۵ است و m,n بیانگر میزان عرض و ارتفاع تصویر می باشند.

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

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

۲-۳- تحلیل نتایج

همانطور که واضح است از لحاظ حجم (تصویر شماره ۸) الگوریتم پیشنهادی ما از ۲ الگوریتم JPG و PNG به مراتب حجم را بیشتر کاهش داده و در قیاس با JPG با کیفیت ۵۰٪ تقریبا برابری دارد، در تصاویری که پیچیدگی کمتری دارند (تعداد رنگ های به کار رفته محدود تر است) از لحاظ حجم فشرده سازی الگوریتم ما نسبتا بهتر عمل می کند.

در تصویر شماره ۹ نمودار میله ای مقایسه بین روش ها در زمان ساخت فایل فشرده سازی شده می باشد، که با توجه به این نمودار متوجه میشویم زمان بیشتری نسبت به الگوریتم های JPG و JPG با کیفیت ۵۰٪ مصرف می کنیم، زمان مصرفی در الگوریتم پیشنهادی ما تقریبا برابر زمان الگوریتم PNG می باشد.

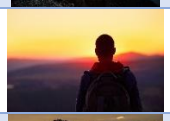
در تصویر شماره ۱۰ مقایسه بین الگوریتم ها در زمان بازیابی تصویر اولیه می باشد، به تعبیر دیگر زمان باز شدن فایل در قالب یک تصویر که به دلیل اینکه روی تک تک پیکسل ها در الگوریتم ما باید عملیات شبه رنگ آمیزی اعمال شود زمان به مراتب بیشتری نسبت به سایر روش ها مصرف می شود.

در تصویر شماره ۱۱ مقایسه معیار سنجش شباهت PSNR می باشد که میزان بیشتر به معنای بهتر بودن است و الگوریتم ما از الگوریتم JPG با کیفیت ۵۰٪ کمی بهتر عمل کرده اما در مقایسه با دو الگوریتم PNG و JPG عملکرد نسبتا ضعیف تری داشته است.

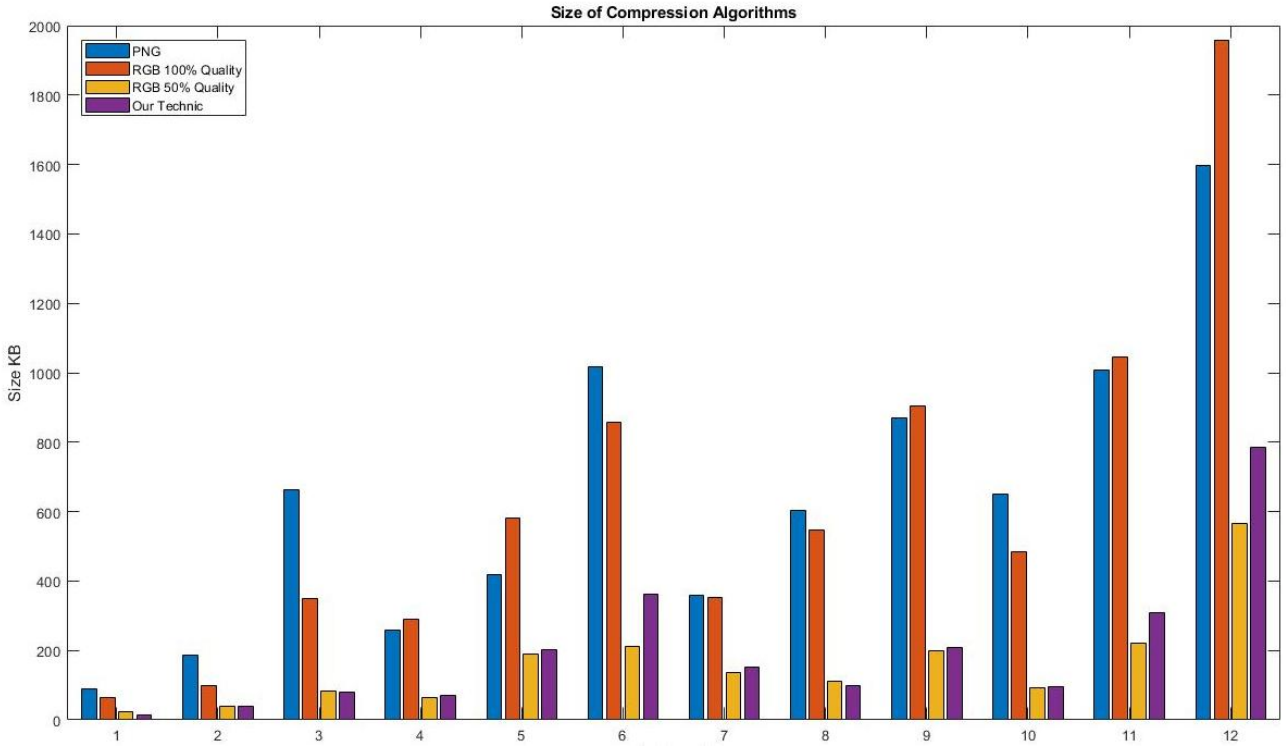
۴- نتیجه گیری و کارهای آینده

در این مقاله از الگوریتم های مبتنی بر محتوا[xxx] برای خوشه بندی استفاده شد که در صورت تعویض با الگوریتم های مبتنی بر چگالی[†††] احتمال می رود نتایج بهتری حاصل شود، در بازیابی تصاویر نیز در صورتی که از منطق فازی استفاده کنیم و یا روی عکس خوشه بندی شده یک فیلتر میانگین گیری انجام دهیم توقع می رود نتایج بهبود یابند.

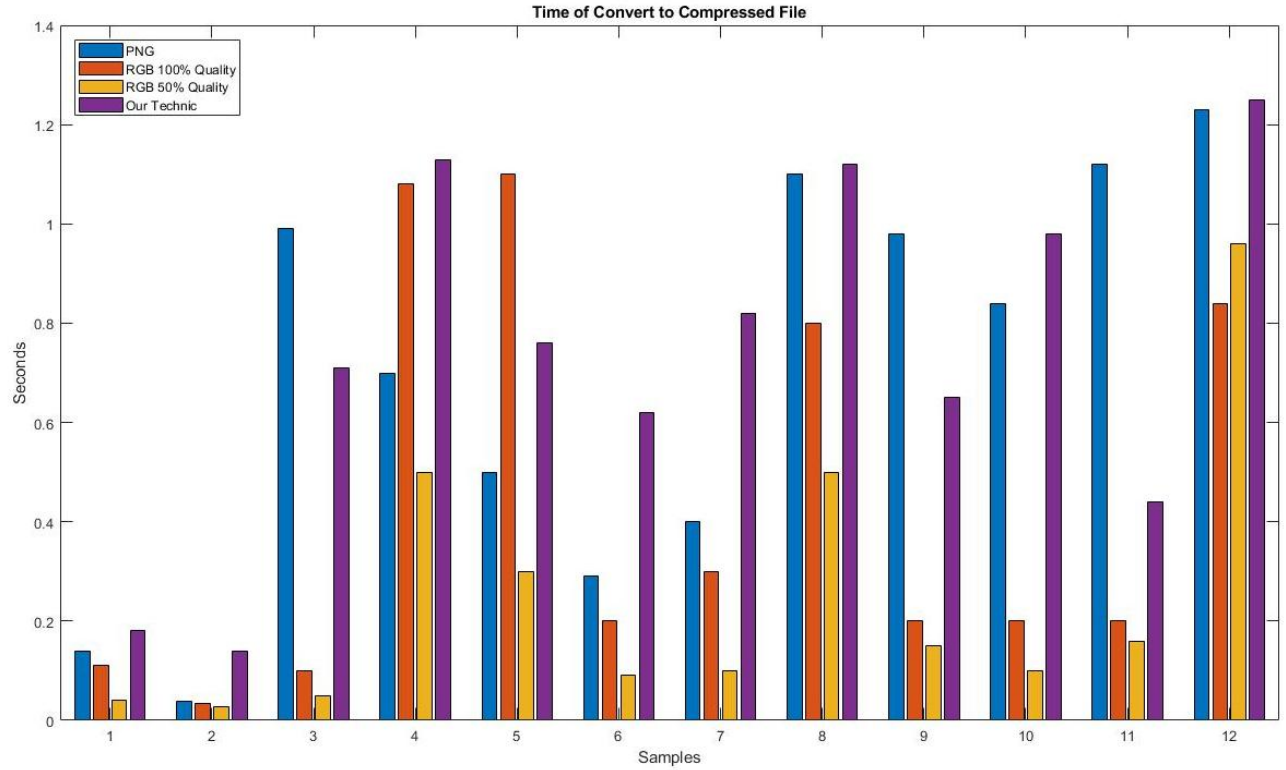
[xxx] Content-Based
[†††] Density-Based

Sample No.	Image	Dimension	size Bitmap
1.		472*296	**412KB**
2.		474*266	**370KB**
3.		960*720	**2026KB**
4.		1500*672	**2954KB**
5.		1200*800	**2813KB**
6.		2048*1365	**8191KB**
7.		1200*702	**2469KB**
8.		2048*1085	**6511KB**
9.		2048*1294	**7.58KB**
10.		2048*1365	**8191KB**
11.		2048*1124	**6745KB**
12.		1536*2048	**9217KB**

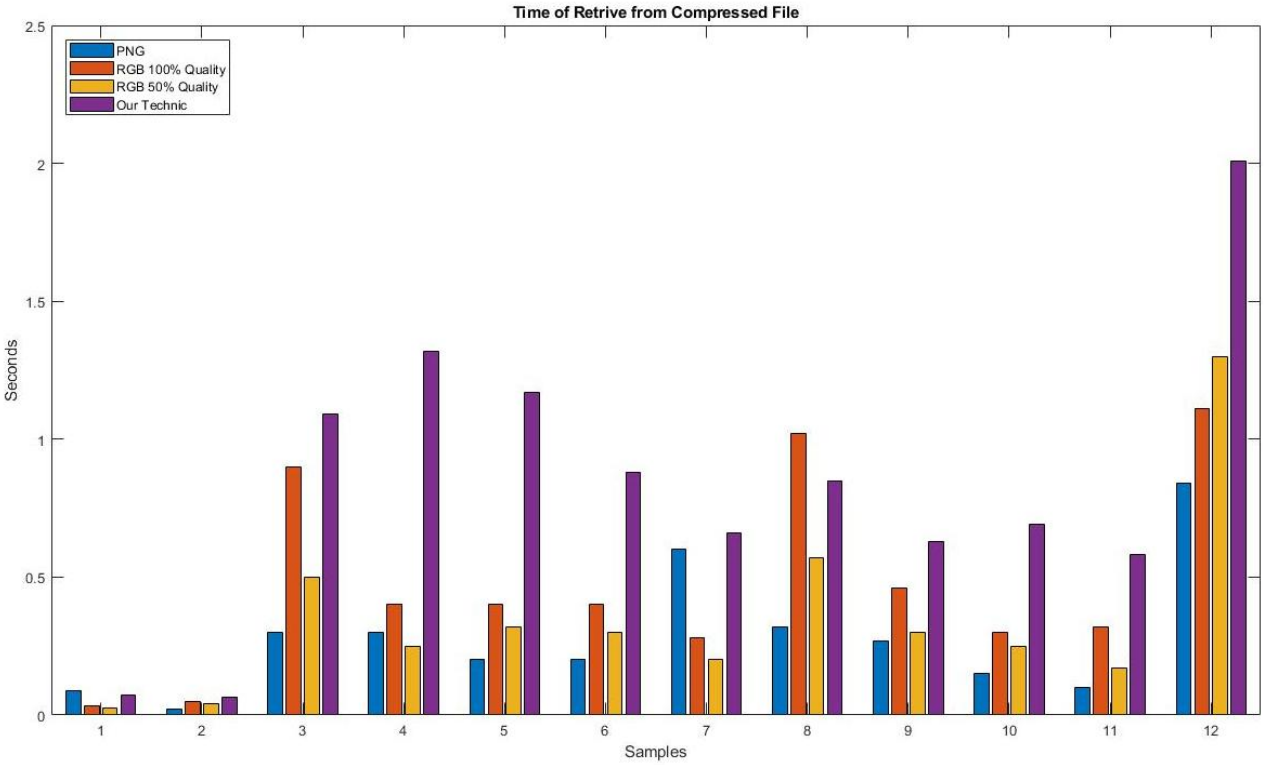
جدول ۱: تصاویر مورد استفاده در نتایج اعمال الگوریتم های مختلف



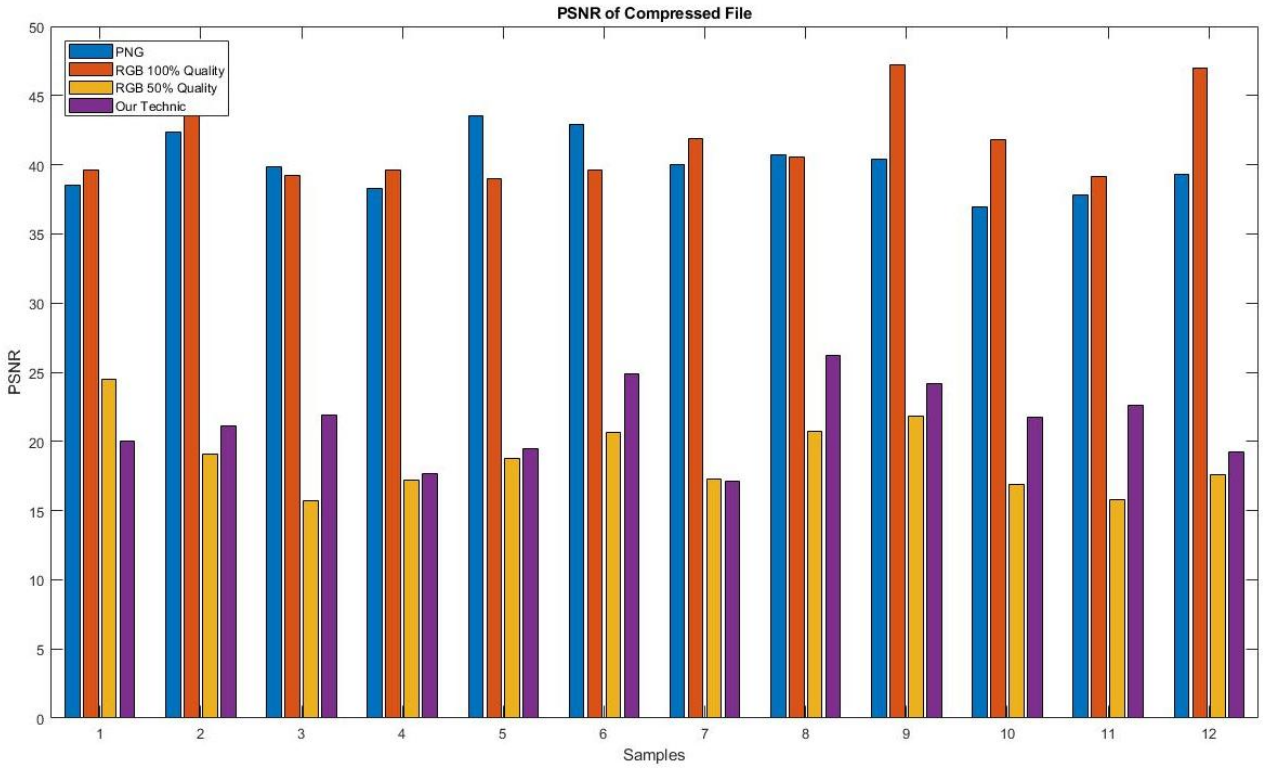
شکل ۸: مقایسه حجم فایل های فشرده سازی شده



شکل ۹: مقایسه زمان ساخت فایل های فشرده سازی شده



شکل ۱۰: مقایسه زمان بازیابی فایل های فشرده سازی شده



شکل ۱۱: مقایسه معیار شباهت PSNR فایل های فشرده سازی شده

مراجع

[1] Image Compression A Survey, Mehwish Rehman and etc. , Research Journal of Applied Sciences, Engineering and Technology 7(4): 656-672, (2014).

[2] A Survey on Parallel Computing of Image Compression Algorithms JPEG and Fractal Image Compression, Priyadarshini K S and etc. , Computational Systems for Health & Sustainability" 17-18 ,April(۲۰۱۵) .

[3] Fractal Images Compressing by Estimating the Closest Neighborhood with Using of Schema Theory, Mahdi Jampour and etc. , Journal of Computer Science 6 (5): 591-596, (2010).

[4] 4. Fast vector quantization using a Bat algorithm for image compression, Chiranjeevi Karri and etc. , Engineering Science and Technology, an International Journal 19 (2016).

[5] 5. A new cluster validity measure and its application to image compression, C.-H. Chou and etc. , Pattern Anal Applic (2004).

[6] Centroidal Voronoi Tessellation Algorithms for Image Compression, Segmentation, and Multichannel Restoration, QIANG DU and etc. , J Math Imaging Vis 24: 177–194(۲۰۰۶) .

[7] Clustering-Based Quantisation for PDE-Based Image Compression, Laurent Hoeltgen and etc. , Mathematics Subject Classification(۲۰۱۰) .

[8] A review of web image mining, Keiji Kanai, ITE Transaction on media Technology and Applications (۲۰۱۵).

[9] Clustering of Image Data Using K-Means and Fuzzy K-Means, Md. Khalid Imam Rahmani and etc. , International Journal of Advanced Computer Science and Applications, Vol. 5, No. 7(۲۰۱۴).

[10] Content-based image retrieval using PSO and k-means clustering algorithm, Zeyad Safaa Younus and etc. , Saudi Society for Geosciences(۲۰۱۴) .

[11] Fast Approximate K-Means via Cluster Closures, Jingdong Wang and etc. , Microsoft(۲۰۱۳) .

[12] EMR A Scalable Graph-based Ranking Model for Content-based Image Retrieval , Bin Xu and etc. , IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY (۲۰۱۵)

[13] A Heuristic Method for Gray Images Pseudo Coloring with Histogram and RGB Layers, Mohammad Zare and etc. , International Conference on Computer and Communication Devices(۲۰۱۱) .

[14] Image Classification and Retrieval are ONE, Lingxi Xie and etc. , ICMR '15, Shanghai, China, June 23 – 26(۲۰۱۵) ,.

[15] Energy efficient distributed image compression in resource-constrained multihop wireless networks, Huaming Wu and etc. , Computer Communications 28 (2005).