



تخمین آستانه با استفاده از برنامه نویسی ژنتیک

جهت نويززدایی در حوزه موجک

ياسر نوروززاده

دانشگاه شهيد باهنر کرمان

nowrouzzadeh@yahoo.com

سراج الدين کاتبی

دانشگاه شیراز

katebi@shirazu.ac.ir

نیست و به تخمین واریانس نویز و یا سایر پارامترها نیازی ندارد.

نتایج آزمایش روش ارائه شده بر روی تصاویر مختلف نشان داد که روش پیشنهادی علاوه بر دقت مناسب، سریعتر از بسیاری از روشهای رایج است و می تواند برای پردازش بلادرنگ تصاویر به کار رود.

واژه های کلیدی: نويززدایی تصاویر، آستانه گیری موجک، تبدیل موجک، برنامه نویسی ژنتیک

1- مقدمه

معمولاً تصاویر دنیای واقعی با نویز همراه هستند. هدف از نويززدایی حذف نویز به صورتی است که ویژگیهای اصلی سیگنال حفظ شوند. در گذشته این عمل با روشهای خطی مانند فیلتر Wiener انجام می گرفت. اخیراً، حجم گسترده ای از کارهای حذف نویز با حضور نویز گوسی جمع شونده و با روشهای غیر خطی انجام می شود. تحقیقات انجام شده در

چکیده: مسأله حذف نویز از داده ها مدتهاست که موضوع تحقیقات در علوم مختلف است. یکی از بخشهای مورد توجه در این زمینه، نويززدایی از تصویر است که کاربردهای فراوانی دارد. در سالهای اخیر با توجه به قدرت تبدیل موجک، روشهای حذف نویز در حوزه موجک مطرح شده اند که نسبت به روشهای قبلی نتایج بهتری را به دنبال داشته اند.

در روشهای حذف نویز در حوزه موجک، باید از مقدار آستانه و تابع آستانه گیری مناسبی استفاده کرد. گروهی از روشهای حذف نویز در حوزه موجک، با استفاده از راه کارهای گوناگون مقدار آستانه را برای حذف ضرائب مربوط به نویز را تخمین می زنند و گروهی دیگر سعی در یافتن تابع آستانه گیری مناسب دارند.

در این مقاله با توجه به توانایی روش برنامه نویسی ژنتیک، رابطه ای برای محاسبه مقدار آستانه برای حذف نویز از تصویر به دست آمده است؛ از آنجا که این رابطه به ویژگیهای تصویر وابسته است، می توان مقدار مناسب آستانه را با حجم محاسباتی اندک به دست آورد. همچنین روش پیشنهادی بر خلاف بسیاری از روشهای موجود به اندازه تصویر وابسته

زمینه حذف نویز از طریق آستانه‌گیری با انقباض موجک¹، نشان می‌دهد که روشهای آستانه‌گیری² موجک دارای نتایج بهتری نسبت به روشهای خطی هستند [1 و 2].

آستانه‌گیری، یک روش غیر خطی است که بر روی ضرائب موجک عمل می‌کند. یکی از روشهای رایج فیلتر کردن نویز گوسی جمع شونده (با میانگین صفر و انحراف معیار S)، از طریق آستانه‌گیری ضرائب موجک است [2]. در این روش هرکدام از ضرائب موجک با یک آستانه مقایسه می‌شوند که اگر قدر مطلق آن کمتر از آستانه باشد، صفر می‌گردد و در غیر این صورت، براساس قانون آستانه‌گیری به کار رفته، نگه داشته می‌شود یا تغییر می‌یابد. در حقیقت نقش آستانه این است که ضرائب کوچک را که اغلب نویز هستند، از ضرائب بزرگ که نشان دهنده ویژگیهای مهم سیگنال می‌باشند، متمایز سازد. معروف‌ترین روشهای آستانه‌گیری، روشهای *VisuShrink* و *SureShrink* هستند. در روش *VisuShrink* مقدار آستانه از $S\sqrt{2\log L}$ به دست می‌آید که S^2 واریانس نویز و L اندازه تصویر می‌باشد؛ مثلاً برای یک تصویر با ابعاد 256×256 مقدار $L=65536$ است. در نتیجه مقدار آستانه بزرگ خواهد بود و موجب می‌شود تصویر مات گردد. روش *SureShrink* شامل ترکیب یک آستانه عمومی با آستانه‌ای که از تخمین زننده مخاطره³ به دست می‌آید، است. این روش برای تصاویر مناسب‌تر از روش *VisuShrink* است.

پس از انتشار مطالعات بنیادی در زمینه نویززدایی، روشهای تخمین آستانه‌گیر موجک وفق‌پذیر با داده توسعه یافته‌اند. تخمین زننده‌هایی بر مبنای روشهای ارزیابی دو قسمتی⁴ برای انتخاب پارامتر آستانه‌گیر توسعه داده شدند [5]؛ در حالی که سایرین آستانه‌گیری را به عنوان یک فرآیند آزمون چند فرضیه مطرح نموده‌اند [5]. اخیراً روشهای آستانه‌گیری به جای آستانه‌گیری از هر یک از ضرائب موجک، بر روی گروهی⁵ از ضرائب موجک به کار رفته‌اند [5]. برخی از این تخمین زننده‌های آستانه‌گیر موجک دارای ویژگیهای تقریباً بهینه

هستند. علاوه بر این، نشان داده شده است که تقریب زننده‌های آستانه‌گیری گروهی موجک دارای مربع میانگین خطای عالی در مقایسه با تقریب زننده‌های آستانه‌گیری جمله به جمله موجک با فرض داشتن نمونه‌های محدود هستند.

از سوی دیگر، بر اساس قانون انتخاب داروین در جانداران، الگوریتم کامپیوتری شبیه‌سازی تکامل زیستی⁶ ارائه شد. این روش برای حل مسائل به عنوان یک روش جستجوی مقاوم استفاده می‌شود. مسائل بهینه‌سازی زیادی هستند که با این الگوریتم‌های الهام گرفته از طبیعت حل شده‌اند. بر اساس این روش، روش برنامه‌نویسی ژنتیک ارائه شد تا برای مسائل مورد استفاده یک برنامه کامپیوتری یا عبارات نمادین به عنوان جواب، بیابد. اخیراً از این روش در مسائل گوناگونی از قبیل دسته‌بندی متون، داده کاوی⁷، تخمین نمادی⁸، شبیه‌سازی سیستمها، کنترل، بهینه‌سازی و زمان‌بندی و پردازش سیگنالها استفاده شده است.

روش برنامه‌نویسی ژنتیک دارای چند قسمت مهم است:

- مجموعه پایانه‌ها⁹: شامل مجموعه‌ای از متغیرها و ثوابت.
- مجموعه توابع: شامل مجموعه‌ای از توابع با حوزه مشخص که همراه با مجموعه پایانه‌ها جوابهای یک مسأله مورد نظر هستند. برای تقریب نمادی، این مجموعه می‌تواند شامل توابع پایه‌ای ریاضی باشد. در حالی که برای مسائل دسته‌بندی، عملگرهای بولی و شرطی به کار می‌روند.
- تابع برازندگی¹⁰: برازندگی مقداری عددی است که به هر عضو یک جمعیت داده می‌شود تا ملاکی برای میزان مناسب بودن یک جواب برای مسأله باشد.

این روش مراحل زیر را برای رسیدن به جوابهای مناسب برای مسأله، طی می‌کند:

1. ایجاد یک مجموعه تصادفی از عبارات (برنامه‌های) نمادین با استفاده از توابع نمادین همراه با پایانه‌های ثابت و نمادین.

⁶ Genetic Algorithm (GA)

⁷ Data mining

⁸ Symbolic regression

⁹ Terminal Set

¹⁰ Fitness function

¹ Wavelet Shrinkage

² Thresholding

³ Risk

⁴ Cross-Validation

⁵ Block

2. ارزیابی هر عبارت تولید شده با مقدار تابع برازندگی از پیش تعریف شده برای بهینه سازی که توانایی عبارت را برای حل مسائل اندازه می گیرد.

3. استفاده از برخی روشهای انتخاب از قبل تعریف شده برای انتخاب برخی والدین برای ترکیب.

4. ترکیب ژنتیکی والدهای انتخاب شده با عملگر تولید مثل برای تولید جمعیت جدید.

5. اعمال عملگر جهش بر روی این جمعیت جدید.

6. تکرار گام 2 تا رسیدن به شرایط پایانی از قبل تعریف شده یا رسیدن به حداکثر تعداد نسلها.

7. جواب مسأله، الگوریتمی است با بهترین مقدار برازندگی در تمام نسلها.

گروهی از روشهای موجود مانند روشهای بیزی، برای کاهش حجم محاسبات، یک توزیع مستقل برای ضرائب موجک تصویر در نظر می گیرند؛ در حالی که این ضرائب به هم و روشهای رایج فعلی بررسی شده اند. در بخش سوم، روش ارائه شده شرح داده شده است. بخش بعد نتایج به دست آمده را در برمی گیرد و در بخش پنج، بحث و نتیجه گیری آمده است.

2- نویززدایی در حوزه موجک

فرض کنید $\{y_{i,j}, i, j=1, \dots, n\}$ تصویر نویزی باشد که با نویز جمع شونده گوسی آسیب دیده است که n توانی از 2 است؛ بنابراین داریم:

$$y_{i,j} = x_{i,j} + se_{i,j} \quad i, j = 1, \dots, n \quad (1)$$

و $\{e_{i,j}\}$ نویز با توزیع نرمال $N(0, s^2)$ و مستقل از $\{x_{i,j}\}$ می باشند. فرض کنید $Y = \{y_{i,j}\}$ ، $X = \{x_{i,j}\}$ و $e = \{e_{i,j}\}$ باشند. هدف، حذف نویز از $\{y_{i,j}\}$ و به دست آوردن یک تقریب $\{\hat{x}_{i,j}\}$ از $\{x_{i,j}\}$ است که مقدار MSE را مینیمم نماید:

$$MSE(\hat{x}) = \frac{1}{n^2} \sum_{i,j=1}^n (\hat{x}_{i,j} - x_{i,j})^2 \quad (2)$$

در روش حذف نویز آستانه گیری موجک، ضرائب زیرباندهای Y فیلتر می شوند تا ضرائب بدون نویز $\{\hat{x}_{i,j}\}$ به دست آیند. سپس با اعمال تبدیل موجک معکوس به آنها، \hat{X} به دست

وابسته هستند. همچنین در بعضی روشها به دلیل نیاز به محاسبه پارامترها یا به دلیل وابستگی به اندازه تصویر، حجم محاسبات افزایش می یابد.

در روشهایی که بر اساس بلوک بندی تصویر کار می کنند، یافتن مقدار مناسب برای طول بلوک مشکل است و همچنین موجب مات شدن تصویر و از بین رفتن لبه ها می شوند.

روش ارائه شده علاوه بر این که مستقل از اندازه تصویر است، حجم محاسباتی اندکی دارد. همچنین به دلیل محاسبه مقدار آستانه بر اساس ویژگیهای زیرباندهای تصویر در حوزه موجک، یک آستانه مبتنی بر مشخصات تصویر به دست می آید که موجب می شود استفاده از این روش برای نویززدایی از تصاویر مختلف، نتایج مناسبی داشته باشد.

این مقاله شامل مطالب زیر است. در بخش دوم، مفهوم نویززدایی در حوزه موجک بیان شده

می آید. دو روش آستانه گیری معمول، آستانه گیری سخت و آستانه گیری نرم هستند که با روابط (3) و (4) بیان می شوند.

$$d_l^H(\hat{d}_{jk}) = \begin{cases} 0 & \text{if } |\hat{d}_{jk}| \leq l \\ \hat{d}_{jk} & \text{if } |\hat{d}_{jk}| > l \end{cases} \quad (3)$$

$$\delta_\lambda^S(\hat{d}_{jk}) = \begin{cases} 0 & \text{if } |\hat{d}_{jk}| \leq \lambda \\ \hat{d}_{jk} - \lambda & \text{if } \hat{d}_{jk} > \lambda \\ \hat{d}_{jk} + \lambda & \text{if } \hat{d}_{jk} < -\lambda \end{cases} \quad (4)$$

که l مقدار آستانه و \hat{d}_{jk} ضرائب موجک تصویر نویزی می باشند. چون روش آستانه گیری نرم دارای نتایج بهتری است، بر آستانه گیری سخت ترجیح داده می شود [1].

گرچه ایده آستانه گیری ساده و مؤثر است، ولی یافتن مقدار مناسب برای آستانه کار آسانی نیست. در روشهای اولیه آستانه عمومی با مقدار $l = s\sqrt{2\log L}$ پیشنهاد شده است [2] که از یک تخمین بهینه از حداقل کردن ماکزیمم خطا بر روی تمام سیگنال به دست آمده است. روش قابل توجه دیگر جهت به دست آوردن مقدار مناسب برای آستانه، روش *SureShrink* است که از می نیمم کردن تخمین زننده ریسک هنگام استفاده از آستانه گیری نرم به دست می آید. آستانه به دست آمده در *VisuShrink* به داده های زیر باندها وابسته است. روش دوم،

دارای کارایی خوبی در حذف نویز از تصویر است و تقریباً کمترین MSE را به دست می آورد [3 و 4]؛ به همین دلیل جهت مقایسه با روش ارائه شده، انتخاب شده است. اخیراً روشهایی برای آستانه‌گیری از گروهها ارائه شده است که به جای آستانه‌گیری هر یک از ضرائب موجک، هر گروه به طور جداگانه آستانه‌گیری می‌شود [5]. در این حالت، اطلاعات بیشتری در مورد هر ضریب خواهیم داشت؛ زیرا ضرائب مجاور هر ضریب نیز در نظر گرفته می‌شوند. در نتیجه آستانه‌بتری تخمین زده می‌شود. در یک روش گروهها، همپوشانی ندارند، ولی در روش دیگر گروهها دارای همپوشانی هستند [5]. همچنین از چارچوب بیز جهت به دست آوردن آستانه وابسته به سطح¹¹ برای گروههای غیرهمپوشان ضرائب موجک، استفاده شده است [5]. این روشها به دلیل دارا بودن اطلاعات بیشتر در مورد هر ضریب، نسبت به روشهای قبلی که آستانه‌گیری جمله به جمله¹² را انجام می‌دهند، دقت بهتری دارند.

3- روش ارائه شده

ایده اصلی روش ارائه شده در این مقاله، در [6] مطرح شد و سپس با استفاده از ANFIS، قوانین فازی برای به دست آوردن آستانه از داده‌های تصویر استخراج شد. در [7] با استفاده از برنامه‌نویسی ژنتیک رابطه‌ای مشابه توابع آستانه‌گیری سخت و نرم به دست آمد؛ با این تفاوت که رابطه به دست آمده نیازی به آستانه برای حذف نویز از تصویر ندارد. نتایج این روش نیز در جداول 1 تا 4 جهت مقایسه آمده است.

در حالت ایده‌آل که تصویر بدون نویز و نویزی را در اختیار داریم، می‌توان آستانه را برای هر زیرباند از رابطه زیر به دست آورد که در آن y_{ij} زیرباند نام در سطح نام از تبدیل موجک گسسته تصویر نویزی است و x_{ij} زیرباند متناظر آن در تصویر سالم است [5].

$$I_k = \sqrt{s_{y_{ij} - x_{ij}}} \quad i, j = 1, \dots, 3 \quad k = 1, \dots, 9 \quad (5)$$

در اینجا جهت فراهم نمودن داده‌های آموزشی برنامه‌نویسی ژنتیک به صورت زیر عمل شده است؛ ابتدا نویزهای گوسی

مختلف در بازه $S = [5:0.1:40]$ به تصویر Barbara اضافه شده است. سپس تبدیل موجک تصویر در سه سطح به دست آمده و میانگین و واریانس زیرباندهای دارای جزئیات، به عنوان ورودی محاسبه شده است. از طرف دیگر با اعمال رابطه (5) به زیرباندهای جزئیات، آستانه ایده‌آل برای هر زیرباند در هر سطح محاسبه شده است که به عنوان خروجی در نظر گرفته می‌شود. مجموعه این داده‌ها برای یافتن عبارتی برای محاسبه آستانه توسط برنامه‌نویسی ژنتیک در مرحله بعد به کاررفته‌اند. شکل 1 روند به دست آوردن داده‌ها را نشان می‌دهد.

عموماً GP نیاز به دو مجموعه توابع و پایانه‌ها برای تولید عبارات دارد. در این کار مجموعه توابع به صورت $\{+, -, /, \times, \sqrt{\quad}\}$ است و ویژگی‌های آماری $\{m_{H_i}, m_{V_i}, m_{D_i}, s_{H_i}, s_{V_i}, s_{D_i}\}$ که $i=1,2,3$ به عنوان مجموعه پایانه‌ها در نظر گرفته شده‌اند و S و m به ترتیب واریانس و میانگین را مشخص می‌کنند و اندیسهای H_i, V_i و D_i به ترتیب زیرباندهای افقی، عمودی و قطری را در همین سطح تبدیل موجک مشخص می‌کنند.

از رابطه (6) به عنوان $RSNR$ استفاده شده است و از آنجا که در حذف نویز، بیشتر بودن مقدار $PSNR$ نشانه بهتر بودن دقت روش مورد استفاده است، تابع برازندگی به صورت رابطه (7) تعریف شده است.

$$PSNR = 20 \times \log_{10} \frac{\max(Denoised)}{\sqrt{MSE}} \quad (6)$$

$$Fitness = MSE \quad (7)$$

در روش ارائه شده تعداد نسلها 1000 بوده و از جمعیت 100 تایی استفاده شده است. روش 2-Point نیز به عنوان روش ترکیب به کار رفته و مقدار احتمال ترکیب و جهش به ترتیب 0/2 و 0/2 بوده‌اند.

4- نتایج: جهت بررسی روش ارائه شده، چند تصویر

متفاوت استفاده شده است که به دلیل تشابه نتایج، در اینجا نتایج آزمایش بر روی سه مورد از آنها آمده‌است. همچنین نویز گوسی با چهار مقدار متفاوت انحراف از معیار، به تصویر

¹¹ Level-Dependent

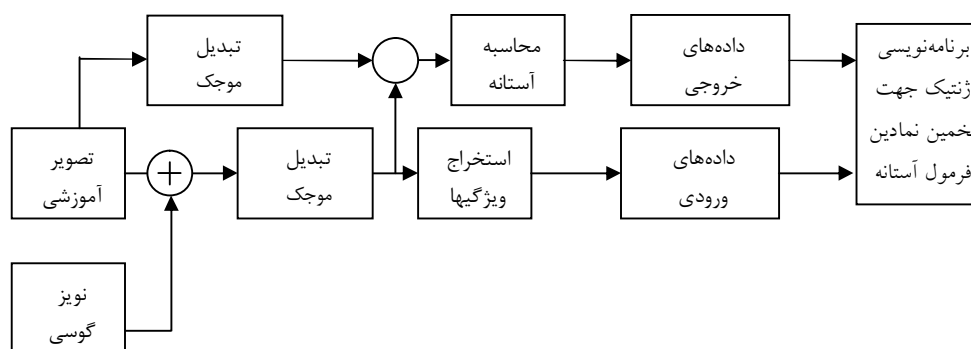
¹² Term-by-Term

جدول 1 تا 4 نتایج مقایسه مقدار $PSNR$ این روش با روشهای دیگر را نشان می‌دهد. تمام مقادیر به‌دست آمده بر اساس میانگین پنج بار اجرای هر روش هستند. بهترین مقدار $PSNR$ در میان روشهای بررسی شده با قلم ضخیم‌تر نمایش داده شده‌اند. همانطور که مشاهده می‌شود، برای نویزهای مختلف در اکثر موارد روش پیشنهادی بهبود خوبی را در مقدار $PSNR$ نشان می‌دهد. شکل 3 نیز مقایسه بین کیفیت تصاویر حذف نویز شده را برای چند روش مختلف نشان می‌دهد. جدول 5 میزان زمان مصرف پردازنده برای روشهای مختلف و روش ارائه شده را بر روی یک کامپیوتر با پردازنده اینتل 2/8

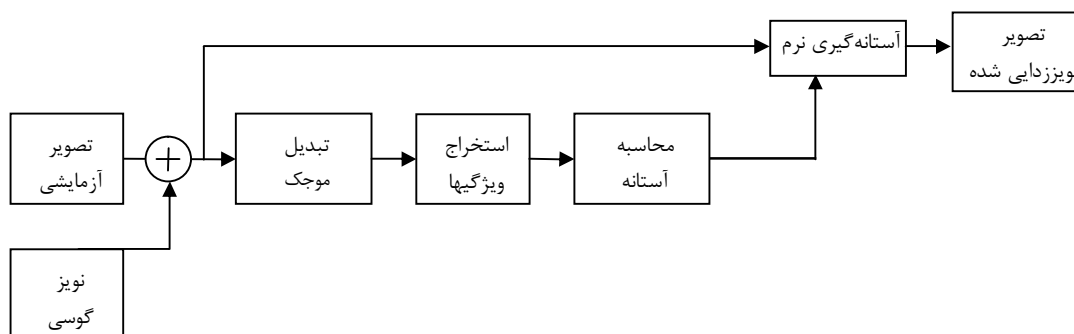
اضافه شده است. هدف به‌دست آوردن رابطه‌ای برای محاسبه مقدار آستانه است به‌طوری که بیشترین مقدار $PSNR$ به‌دست آید. عبارت بهینه محاسبه شده توسط برنامه‌نویسی ژنتیک برای تعیین مقدار آستانه در رابطه (8) مشخص شده است.

$$\begin{aligned} a &= s_{V_2} - 4s_{D_1} + 10s_{D_3} + 3s_{H_3} + 4\sqrt{s_{D_2}} \\ b &= \sqrt{s_{V_2} - 3s_{D_1} + 2s_{H_3} + 6s_{D_3}} + 4\sqrt{s_{D_2}} \\ I &= 4\sqrt{ab} \end{aligned} \quad (8)$$

سپس با استفاده از آستانه محاسبه شده از طریق رابطه (8)، و با اعمال قانون آستانه‌گیری نرم، عمل نویززدایی انجام شده است. شکل 2 نحوه استفاده از آستانه محاسبه شده با رابطه (8) جهت حذف نویز از تصویر را نشان می‌دهد.



شکل (1) نحوه فراهم کردن داده‌ها برای تخمین آستانه‌ها توسط برنامه‌نویسی ژنتیک



شکل (2) نحوه آزمایش آستانه‌های تخمین‌زده شده توسط برنامه‌نویسی ژنتیک

5- بحث و نتیجه‌گیری: روشهای موجود را می‌توان از دو جنبه مقایسه کرد. یکی از نظر مقدار $PSNR$ و دیگری از نظر حجم محاسباتی. با توجه به جداول فوق ملاحظه می‌شود که روشهایی که دقت مناسبی دارند (مانند CV و $Bayes$) دارای

گیگاهرتز و حافظه 512 مگابایت را نشان می‌دهد. برای سایر تصاویر مورد استفاده نیز مقادیر مشابهی به‌دست آمده است.

مزیت دیگر روش ارائه شده در این مقاله این است که یک فرمول وفقی پیشنهاد می‌کند که بر اساس ویژگیهای مؤثر و ساده تصویر بیان شده است و در عین حال به ابعاد تصویر بستگی ندارد. در نتیجه برای تصاویر مختلف می‌توان همین نتایج را انتظار داشت.

به نظر می‌رسد با تغییر مجموعه توابع و مجموعه پایانه‌ها در برنامه‌نویسی ژنتیک بتوان به روابط جدیدی دست یافت که مقدار $PSNR$ را بهبود دهند. اگر تابع برازندگی را به گونه دیگری تعریف کنیم، می‌توان انتظار داشت که رابطه دقیق‌تری برای محاسبه مقدار آستانه به‌دست آید.

حجم محاسباتی بسیار بالایی در مقایسه با روشهای دیگر دارند. به عبارت دیگر در روشهای موجود میزان دقت روش با حجم محاسباتی آنها رابطه مستقیم دارد و برای به‌دست آوردن دقت بالاتر، زمان بیشتری از پردازنده مصرف می‌شود. دلیل این امر آن است که این روشها نیاز به محاسبه پارامترهایی زیادی دارند که طبعاً محاسبه آنها زمان‌گیر است؛ در حالی که روش پیشنهادی هر دو مزیت را با هم داراست؛ یعنی ضمن این که قدرت نویززدایی مناسبی دارد، حجم محاسباتی کمی را نیز دربر دارد. بنابراین می‌تواند در عمل کاربردهای بیشتری نسبت به روشهای موجود داشته باشد.

با توجه به مشکلات روشهای موجود، در [6 و 7] تخمین آستانه به صورت یک مسأله بهینه سازی مطرح شد و سپس با استفاده از روشهای هوش مصنوعی این ایده پیاده‌سازی شد. در [6] هدف یافتن مقدار مناسب برای آستانه بود و در [7] هدف یافتن یک تابع آستانه‌گیری مناسب بود.



(الف)



(ب)



(ج)



(د)

شکل (3) مقایسه تصاویر نویز زدایی شده (الف) تصویر Lena با $S = 35$ (ب) CV (ج) $BayesBlockMean$ (د) روش ارائه شده

جدول ۱ مقایسه روش پیشنهادی با روشهای دیگر از نظر مقدار PSNR

برای تصویر Baboon

| Method | PSNR | | | |
|--------------------------|--------------|--------------|--------------|--------------|
| | $s = 10$ | $s = 20$ | $s = 30$ | $s = 35$ |
| VisuShrink | 19.11 | 18.46 | 17.62 | 17.19 |
| HybridSure | 26.54 | 23.47 | 20.94 | 19.86 |
| Wiener | 23.10 | 21.80 | 20.26 | 19.49 |
| NeightBlock | 20.10 | 18.86 | 17.61 | 17.03 |
| CV | 31.41 | 25.53 | 22.13 | 20.86 |
| BayesBlockMean | 27.49 | 24.31 | 21.67 | 20.56 |
| GP Thresholding function | 30.56 | 25.32 | 21.96 | 20.66 |
| Proposed Method | 31.65 | 25.64 | 22.10 | 20.78 |

جدول ۲ مقایسه روش پیشنهادی با روشهای دیگر از نظر مقدار PSNR

برای تصویر Barbara

| Method | PSNR | | | |
|--------------------------|--------------|--------------|--------------|--------------|
| | $s = 10$ | $s = 20$ | $s = 30$ | $s = 35$ |
| VisuShrink | 26.52 | 23.26 | 20.92 | 19.95 |
| HybridSure | 31.10 | 25.53 | 21.97 | 20.77 |
| Wiener | 28.66 | 25.05 | 22.19 | 21.02 |
| NeightBlock | 29.31 | 24.42 | 21.40 | 20.24 |
| CV | 31.41 | 25.53 | 22.13 | 20.86 |
| BayesBlockMean | 31.18 | 25.67 | 22.34 | 21.06 |
| GP Thresholding function | 29.53 | 24.83 | 21.60 | 20.33 |
| Proposed Method | 31.38 | 25.35 | 21.84 | 20.51 |

جدول ۳ مقایسه روش پیشنهادی با روشهای دیگر از نظر مقدار PSNR

برای تصویر Goldhill

| Method | PSNR | | | |
|--------------------------|--------------|--------------|--------------|--------------|
| | $s = 10$ | $s = 20$ | $s = 30$ | $s = 35$ |
| VisuShrink | 24.54 | 22.72 | 20.94 | 20.14 |
| HybridSure | 30.18 | 25.73 | 22.47 | 21.25 |
| Wiener | 28.50 | 25.46 | 22.83 | 21.72 |
| NeightBlock | 26.14 | 23.43 | 21.20 | 20.26 |
| CV | 31.51 | 26.07 | 22.76 | 21.51 |
| BayesBlockMean | 31.25 | 26.17 | 22.93 | 21.36 |
| GP Thresholding function | 30.88 | 25.81 | 22.51 | 21.23 |
| Proposed Method | 32.26 | 26.24 | 22.72 | 21.38 |

جدول ۴ مقایسه روش پیشنهادی با روشهای دیگر از نظر مقدار PSNR

برای تصویر Lena

| Method | PSNR | | | |
|--------------------------|--------------|--------------|--------------|--------------|
| | $s = 10$ | $s = 20$ | $s = 30$ | $s = 35$ |
| VisuShrink | 26.39 | 22.78 | 20.35 | 19.37 |
| HybridSure | 32.14 | 26.56 | 23.09 | 21.85 |
| Wiener | 30.66 | 26.43 | 23.35 | 22.10 |
| NeightBlock | 30.85 | 25.72 | 22.56 | 21.35 |
| CV | 31.51 | 26.07 | 22.76 | 21.51 |
| BayesBlockMean | 32.24 | 26.62 | 23.12 | 21.89 |
| GP Thresholding function | 31.03 | 25.93 | 22.56 | 21.25 |
| Proposed Method | 32.30 | 26.28 | 22.75 | 21.42 |

جدول ۵ مقایسه روش پیشنهادی با روشهای دیگر از نظر مقدار زمان بر

حسب ثانیه پردازنده برای تصویر Baboon

| Method | CPU Time in seconds | | | |
|--------------------------|---------------------|----------|----------|----------|
| | $s = 10$ | $s = 20$ | $s = 30$ | $s = 35$ |
| VisuShrink | 0.11 | 0.11 | 0.11 | 0.11 |
| HybridSure | 0.14 | 0.15 | 0.13 | 0.13 |
| Wiener | 0.14 | 0.12 | 0.14 | 0.12 |
| NeightBlock | 0.31 | 0.31 | 0.31 | 0.31 |
| CV | 8.46 | 8.54 | 8.40 | 8.61 |
| BayesBlockMean | 588.84 | 568.78 | 574.26 | 555.46 |
| GP Thresholding function | 0.23 | 0.22 | 0.22 | 0.22 |
| Proposed Method | 0.24 | 0.23 | 0.23 | 0.22 |

مراجع

- [1] D. L. Donoho, "De-noising by soft-thresholding," IEEE Trans. Inform. Theory, vol. 41, pp. 613–627, May 1995.
- [2] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," Biometrika, vol. 81, pp. 425–455, 1994.
- [3] R. A. DeVore and B. J. Lucier, "Fast wavelet techniques for near-optimal image processing," in IEEE Military Communications Conf. Rec. San Diego, Oct. 11–14, vol. 3, pp. 1129–1135, 1992.
- [4] A. Chambolle, R. A. DeVore, N. Lee, and B. J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," IEEE Trans. Image Processing, vol. 7, pp. 319–335, 1998.
- [5] A. Antoniadis, J. Bigot and T. Sapatinas. "Wavelet estimators in nonparametric regression: a comparative simulation study," Journal of Statistical Software, Vol. 6, Issue 6, 1-83, 2001.
- [6] Y. Norouzzadeh, S.D Katebi, "Application of ANFIS in Wavelet Denoising," 6th Iranian Conference on Fuzzy Systems and 1th Islamic World Conference on Fuzzy Systems, Islamic Azad University of Shiraz, 2006.
- [7] مهدی افشاری، یاسر نوروززاده، سراج الدین کاتبی، "حذف نویز از تصاویر رقمی با استفاده از برنامه نویسی ژنتیک و تبدیل موجک،" مجموعه مقالات هفتمین کنفرانس سیستمهای هوشمند، دانشگاه خواجه نصیرالدین طوسی، 1384